Masterthesis:
Immunohistochemical Image Generation using Deep Learning

written by
Philipp Rosin (338584)

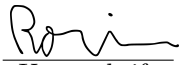
First Supervisor : Prof. Dr. O. Hellwich
Second Supervisor: Prof. Dr. Marc Alexa

Technische Universität Berlin
Fakultät IV
Fachgebiet Computer Vision & Remote Sensing

Contents

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, 6.2.2024 

Ort, Datum Unterschrift

Danksagung

Ich danke Monika und Simon, die immer ein offenes Ohr für Fragen hatten und mir mit guten Ratschlägen beistanden.
Anke und Ralf, deren bedingungslose Unterstützung mir diesen akademischen Weg ermöglicht hat.
Larissa und Philippe deren Freundschaft über die Zeit meines Studiums und während dieser Arbeit immer eine Quelle der Kraft waren.

1 Abstract

This master thesis presents a comprehensive survey comparing ten distinct deep learning methods employed for the paired image translation task of generating immunohistochemistry (IHC)-stained images from their hematoxylin and eosin (HE)-stained counterparts. The investigated architectures encompass convolutional U-Nets, vision transformers, Pix2pix GANs, and a Diffusion Model. The approaches were evaluated quantitatively with metrics such as Mean Squared Error (MSE), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) and qualitatively through visual inspection and the performance of a classifier to predict the IHC-scores of the generated images.
Overall, large U-Nets provide the best solution for this task while transformer based architectures struggle with pixel accuracy while the GAN framework only seems to have an effect on the generated images. This research aims to provide insights into the efficacy and performance variations among these diverse deep learning approaches, shedding light on their potential applications in the vital domain of medical image translation for enhanced diagnostic processes.

Kurzfassung

Diese Masterarbeit präsentiert eine umfassende Übersicht, in der zehn verschiedene Deep-Learning-Methoden verglichen werden, die für die Aufgabe der gepaarten Bildübersetzung eingesetzt werden, um immunhistochemisch (IHC) gefärbte Bilder aus ihren mit Hämatoxylin und Eosin (HE) gefärbten Gegenstücken zu erzeugen. Die untersuchten Architekturen umfassen Faltungs-U-Nets, Vision Transformer, Pix2pix GANs und ein Diffusionsmodell. Die Ansätze wurden quantitativ mit Metriken wie Mean Squared Error (MSE), Structural Similarity Index (SSIM) und Peak Signal-to-Noise Ratio (PSNR) sowie qualitativ durch visuelle Inspektion. Zusätzlich wurde ein Klassifikators zur Vorhersage des IHC-scores implementiert welcher die Netzwerke bewerten soll anhand der Leistung auf den generierten Bilder.
Insgesamt bieten große U-Nets die beste Lösung für diese Aufgabe, während transformatorbasierte Architekturen mit der Pixelgenauigkeit zu kämpfen haben und das GAN-Framework nur einen Einfluss auf die generierten Bilder zu haben scheint. Ziel dieser Forschung ist es, Einblicke in die Wirksamkeits- und Leistungsunterschiede zwischen diesen verschiedenen Deep-Learning-Ansätzen zu gewinnen und Licht auf ihre potenziellen Anwendungen im wichtigen Bereich der medizinischen Bildübersetzung für verbesserte Diagnoseprozesse zu werfen.

2 Motivation

Breast cancer stands as a leading contributor to female mortality. The conventional approach for identifying breast cancer involves histopathological examination, considered the gold standard. This entails creating Hematoxylin and Eosin (HE) stained slices from tumor materials with pathologists conducting diagnoses by either visually inspecting the HE slices through a microscope or analyzing digitized whole slide images (WSI).
For effectively managing diagnosed cases of breast cancer, it becomes imperative to devise precise treatment strategies by assessing the expression of specific proteins, such as human epidermal growth factor receptor 2 (HER2). The standard procedure for evaluating HER2 expression involves immunohistochemical techniques (IHC).
However, there are certain limitations in assessing the level of HER2 expression levels through IHC technology: The creation of IHC-stained slices is a costly process and tumors exhibit heterogeneity, yet in clinical applications, IHC staining is typically conducted on just one pathological slice, which may not entirely reflect the status of the tumor.
Therefore, the development of a method for directly generating IHC images based on HE images might be hugely beneficial for breast cancer patients worldwide. This approach not only reduces the expense associated with IHC-staining but also allows for the generation of IHC images from multiple pathological tissues within the same patient, facilitating a more comprehensive assessment of HER2 expression levels.

3 Medical Background

In medical diagnostics and pathology, two fundamental staining techniques, Hematoxylin and Eosin (HE) staining and Immunohistochemistry (IHC) staining, play pivotal roles. While both techniques contribute to our understanding of tissue samples, they do so in markedly different ways, serving distinct purposes within the field.

HE staining is a cornerstone of pathology, offering a view of tissue structure. It provides a macroscopic assessment, revealing cell types, structural abnormalities, and morphological changes. HE staining is essential for diagnosing diseases, identifying anomalies, and assessing the extent of tissue damage. Hematoxylin stains cellular nuclei purple-blue, while Eosin stains cytoplasm and other cellular structures pink or red. It provides a broad overview of the tissue's cellular composition and architecture but does not target specific proteins.

In IHC staining, antibodies specific to the target protein are used. The secondary antibodies are linked to a reporter molecule (e.g.,an enzyme or fluorophore) that produces a signal at the location of the target protein. This signal allows for the precise localization of the protein of interest within the tissue. IHC staining is ubiquitously used for studying molecular events and protein expression in both research and diagnostic settings.

3.1 Hematoxylin and Eosin staining process

The HE stain is a widely used histological staining technique that plays a crucial role in the visualization and examination of tissue samples. This staining method allows for the differentiation of various tissue components and provides valuable information about cellular structure and morphology.

The H&E staining process involves several key steps:

- **Tissue Sectioning and Mounting:** Initially, tissue samples are embedded in paraffin wax, sliced into thin sections, and mounted onto glass slides. These sections need to be thin enough to allow for microscopic examination.
- **Hematoxylin Staining:** Hematoxylin, a basic dye, is the first step in the staining process. Hematoxylin stains cellular nuclei and other basophilic structures blue-purple. This step involves immersing the tissue sections in a Hematoxylin solution, allowing the dye to bind to the DNA located in the nucleus.
- **Differentiation:** After Hematoxylin staining, the excess dye is washed away, and a differentiation step follows. This step typically involves immersing the sections in a weak acid or alcohol solution. The differentiation step is crucial because it helps control the intensity of staining and ensures that cellular details are visible.
- **Eosin Counterstaining:** Following differentiation, the tissue sections are stained with Eosin, an acidic dye. Eosin stains cytoplasm, extracellular matrix, and other acidophilic structures pink or red. The eosinophilic structures contrast with the basophilic nuclei stained by Hematoxylin, allowing for the visualization of cellular components and tissue architecture.
- **Dehydration and Mounting:** After Eosin staining, the tissue sections are dehydrated using a series of alcohol solutions. Finally, the dehydrated sections are cleared in xylene and mounted with a coverslip using a mounting medium. This step is necessary to prepare the slides for microscopic examination.

H&E staining is a fundamental technique in histology and pathology, providing insights into tissue structure and helping to identify abnormalities, such as cellular changes in diseases or tumors. The combined use of Hematoxylin and Eosin allows for the visualization of different tissue components and is essential for the interpretation of histological specimens in various fields, including medicine, biology, and research.

3.2 Immunohistochemical staining

Immunohistochemical staining (IHC) is a widely employed technique in the fields of histopathology and biomedical research. It enables researchers and pathologists to investigate the distribution and localization of specific proteins and biomarkers within biological tissues at the microscopic level. Originally introduced by Albert Coons in 1941[1], IHC has since become an indispensable tool in modern life sciences, aiding in understanding cell morphology, tissue architecture, and the pathogenesis of various diseases, including cancerous tumors.
The examination of tissue suing immunohistochemistry (IHC) involves a series of systematic steps aimed at detecting specific antigens while preserving their cellular context. This process begins with tissue fixation, a critical step for maintaining the structural integrity of the specimen and preventing postmortem protein degradation. The choice of fixation method, such as formalin fixation, can impact antigen preservation and accessibility, with different fixatives used based on the specific antigens under study.

The ICH staining process involves several key steps:

- **Tissue Preparation:** Biological tissues are typically prepared by embedding them in paraffin or freezing them in optimal cutting temperature (OCT) compound. These tissues are then sliced into thin sections, which are subsequently mounted onto glass slides.
- **Antigen Retrieval:** To unmask epitopes and enhance antibody binding, antigen retrieval techniques are employed, especially for intracellular targets. In some cases, tissue sections are permeabilized to facilitate antibody access to intracellular targets, a crucial step for the detection of intracellular antigens.
- **Blocking:** To minimize nonspecific binding, tissue sections are treated with blocking agents like bovine serum albumin (BSA) and normal animal sera. These agents saturate unoccupied binding sites, on the tissue, reducing background staining.
- **Primary Antibody Application:** The primary antibody, meticulously selected to recognize a specific antigen of interest, is applied to the tissue sections. The interaction between the primary antibody and the target antigen anchors the antibody to the protein of interest.
- **Secondary Antibody Application:** Following a thorough washing step to remove unbound primary antibodies, a secondary antibody is applied. The secondary antibody is designed to recognize and bind to the primary antibody, amplifying the signal and facilitating the detection process. Secondary antibodies are usually conjugated with labels such as enzymes (e.g., horseradish peroxidase) or fluorophores, allowing signal detection and visualization.

- **Visualization:** In enzyme-linked detection, a chromogenic substrate is applied, leading to the formation of a visible colored precipitate at the site of antigen-antibody interaction. In contrast, fluorophore-linked secondary antibodies are employed in fluorescence microscopy, offering real-time imaging of protein localization.

These sequential steps in IHC are fundamental for the precise and specific detection of antigens within tissue samples, making it a powerful technique in the field of biomedical research and diagnostics. Researchers should carefully select the appropriate protocols and reagents tailored to their specific experiments, as the details and reagents can vary based on the experimental setup and the antibodies used.

3.3 IHC scoring

Immunohistochemistry (IHC) scores, often represented as 0, 1+, 2+, and 3+, are used to semiquantitatively assess the expression or intensity of a specific antigen or protein in tissue samples. These scores help pathologists and researchers describe the staining pattern and intensity observed in IHC-stained sections. The following is an overview of what each score typically represents:

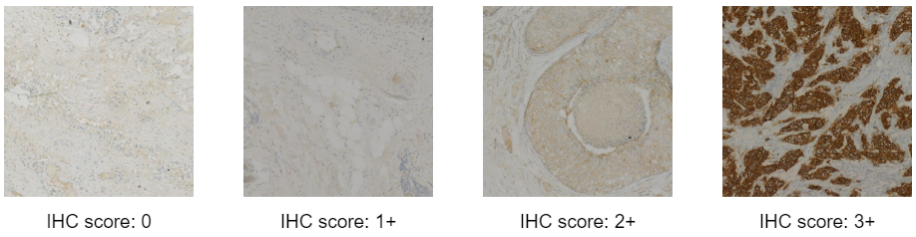


Figure 1: IHC scoring examples

- **0 (Negative):** No staining or minimal staining is observed. This is normally considered as a negative or non-expressive result.
- **1+ (Weak):** Weak or faint staining is observed. This is often considered as weakly positive result but may not have significant biological or clinical relevance.
- **2+ (Moderate):** The sample exhibits moderate staining intensity. This is typically considered as a positive result with some clinical significance.
- **3+ (Strong):** Strong, intense staining is observed. This is often considered as strongly positive result, with potential clinical significance, particularly in the context of disease diagnosis or prognosis.

It is important to note that the interpretation of IHC scores may vary depending on the specific protein being analyzed, the context of the study, and the established criteria for scoring in a given laboratory or clinical setting. Different proteins may have different baseline levels of expression in normal tissue, so what constitutes "weak" or "strong" staining can be relative to the expected expression of that specific protein.

In practice, pathologists and researchers assign these scores based on visual assessment, and the scores may be used to guide clinical decisions, such as cancer grading, prognosis, or treatment selection. To ensure consistency and accuracy, standardized scoring systems and guidelines are often established for specific proteins or diseases.

3.4 HER2 test

The HER2 (Human Epidermal Growth Factor Receptor 2) test is a critical component in the assessment of certain cancers, particularly breast cancer. It involves the evaluation of the HER2 protein expression, which is a key factor in the prognosis and treatment planning for patients. The two primary methods employed in HER2 testing are Immunohistochemistry (IHC) 3.3 scoring and Fluorescence In Situ Hybridization (FISH) testing. Complementing the IHC score, the FISH test provides a molecular analysis of the HER2 gene amplification. This test involves labeling the HER2 gene with fluorescent probes to determine its quantity within the cancer cells. A positive FISH result confirms HER2 gene amplification and further supports the positive IHC findings. FISH testing is particularly valuable in cases where the IHC score is equivocal (2+) or when a more precise determination of the HER2 gene status is needed.

Together, the IHC score and FISH test play a pivotal role in guiding therapeutic decisions for patients with HER2-positive breast cancer. A positive HER2 status often indicates eligibility for targeted therapies, such as HER2 inhibitors, which have demonstrated significant efficacy in improving outcomes for patients with HER2-positive breast cancer. These tests are integral components of a comprehensive diagnostic approach, aiding clinicians in tailoring treatment strategies to the specific molecular characteristics of the patient's cancer.

4 Computer Vision Background

Translating the HE-stained images into their IHC-stained counterparts is a task which falls in the domain of computer vision under the umbrella of paired image translation. Paired image translation describes the task to transform an image between two domains in the field of computer vision. The image pairs have a one to one correspondence so that the pair is not randomly sampled from the target domain but fits uniquely to the image from the source domain. Classic examples for paired image translation include image segmentation or translating a photography from day to night time lighting. In this work the HE-stained images form the source domain and the IHC-stained images form the target domain. In the following chapter different architectures that have proven useful in similar paired image translation tasks are discussed.

4.1 U-Net

The U-Net Architecture is a convectional neural network that was developed at the University of Freiburg to perform biomedical image segmentation [10]. It consists of an encoding path a bottleneck and a decoding path. This U-shaped layout helps to capture both local and global contextual information.

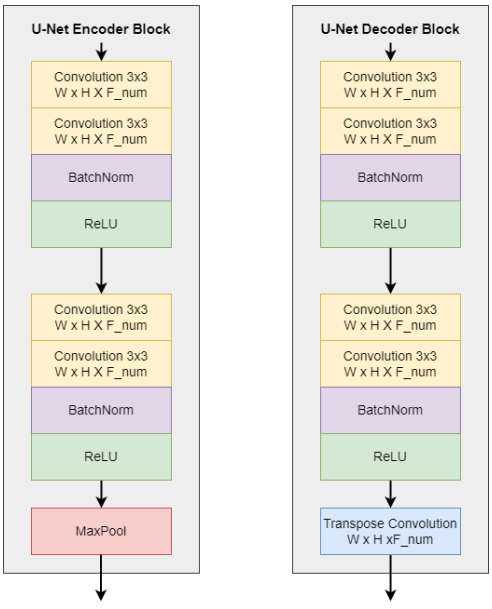


Figure 2: U-Net encoder and decoder blocks

The encoder blocks consist of two convolutional blocks which in turn are made of two convolutional layers with a batchnorm and a ReLu layer. The

encoder block lastly contains a maxpooling layer of kernel size = 2 and stride = 2 so that the image height and width are reduced by half as shown in 4.1. The decoder blocks are made of the same convolutional blocks but a transpose convolution is added so that the image can be up-sampled after passing the bottleneck.

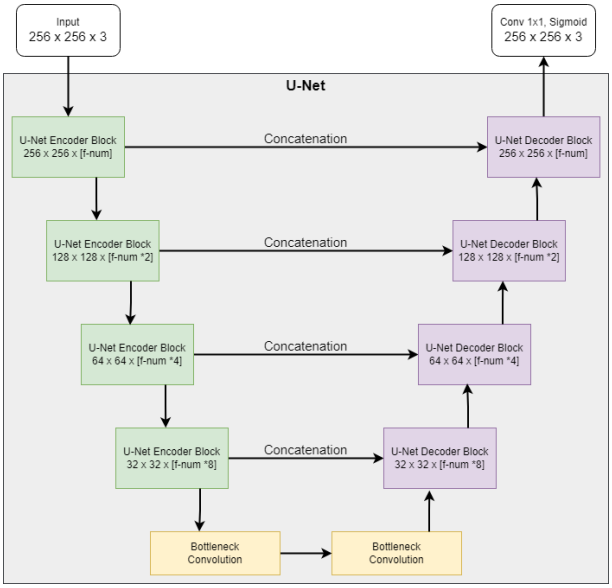


Figure 3: U-net architecture

In the flowchart above, the whole architecture of the U-Net is shown, here four encoder and decoder blocks are used. These pairings are also known as steps. Each step down halves the height and width of the images due to the maxpooling and doubles the channel dimension in the convolution. The reduction in the spacial dimension helps the network to capture local and global features progressively. In the bottleneck the compressed image is further edited by convolution while the dimensions remain the same. Finally the feature map is up-sampled to the original image size by the decoder. The U-net uses skip connections between the encoder and decoder for each step. By introducing these skip connections, the model can merge high-resolution information from the encoder with low-resolution contextual information from the decoder, which helps in accurate segmentation. To train the network the Mean Squared Error (L2-loss) is used as loss function :

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

4.2 Vision Transformers

The Vision Transformer is an architecture introduced by Alexey Dosovitskiy and his colleagues from Google Research originally for image classification [3]. ViTs apply the transformer architecture [12] originally developed for natural language processing. Later it was adapted to the domain of computer vision.

The Vision Transformer treats an image as a sequence of fixed-size non-overlapping patches. These patches are than linearly flattened into a 1D sequence of feature vectors. This sequence of patches serves as the input to the Transformer. To maintain the spatial relationships in the input sequence positional embedding is added to the patch embedding.

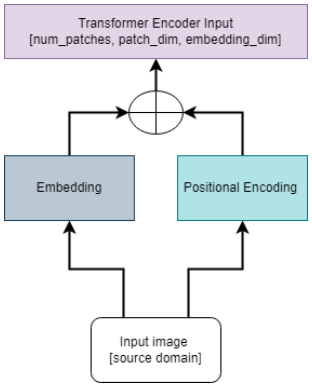


Figure 4: Patch embedding and Transformer input

The Transformer encoder consists of multiple Transformer Blocks with self-attention and feed-forward neural network layers. Self-attention is to compute the importance or relevance of each element in a sequence with respect to all other elements in the same sequence, capturing global context and dependencies in the data. For each element in the sequence, three vectors are derived: the Key K vector, the Query Q vector, and the Value V vector. These vectors are calculated by linear projections which are learned by the network. The Key vector represents the information that will be used to compute the attention score. The Query vector represents the element whose relevance we want to compute, and the Value vector is used to compute the weighted sum of values based on the attention scores.

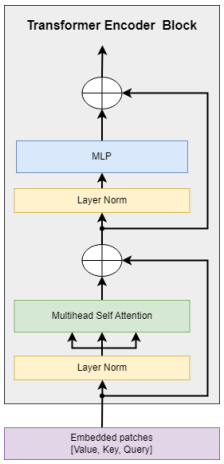


Figure 5: Transformer Block flowchart

After processing the patches through the Transformer blocks the resulting embedding is passed through a classification head in the original implementation. But the Vision Transformer can also be used for image to image translation tasks like segmentation [13]. For the image to image translation the classification head is replaced with an operation that rearranges the image patches back to the size of the original image.

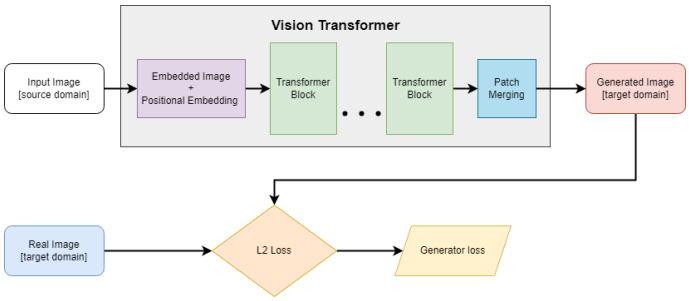


Figure 6: Transformer flowchart

The ViT is trained with an L2-loss 1 between the generated image and the real image of the target domain.

4.3 Swin Transformer

The Swin Transformer [9] is a descendent of the ViT [3]. Here as well the original image is divided into patches and a positional embedding is added but instead

of the computationally expensive self attention the Swin Transformer introduces sliding windows. Instead of attending to all patches, the model attends to a fixed-size window of patches at a time. This brings down the computational effort which is proportional to the suqare of the sequence length in the self-attention for the original ViT. While the computational effort in the Swin Transformer is only proportional to the number of patches in the window time the sequence length.

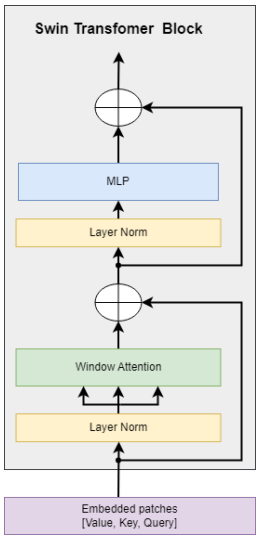


Figure 7: Swin Transformer block

The window is shifted over the patches to capture long-range dependencies effectively. The Swin Transformer employs a hierarchical design organizing multiple Transformer layers into stages.

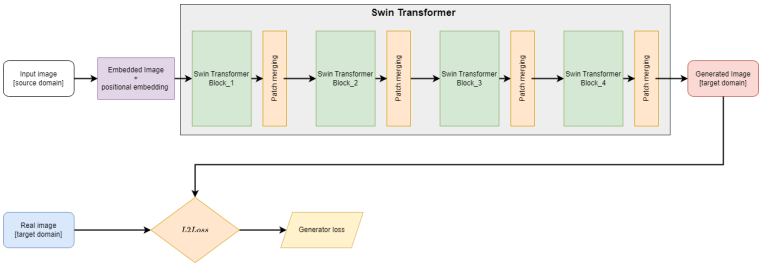


Figure 8: Swin Transfomer flowchart

Each stage consists of a Swin block 4.3 and a patch-merging layer. In the

Swin Transformer paper [9] the progressive stages process the patch embedding at different scales, allowing the model to capture both local and global context efficiently. For the paired image task this is not necessary as the output is also an image of the same size and not a classification.

4.4 Conditional Adversarial Networks

4.4.1 GAN-architecture

Generative Adversarial Networks (GAN) is a framework for image generation with deep neuronal networks that was originally introduced by Ian Goodfellow and his colleagues [4]. Therefore, 2 nets are trained adversarially. The generator tries to generate a realistic version of an image in the source domain either from an image out of a source domain or just noise. In the second step the discriminator receives either a real or generated instance of the target domain or image in the source domain and tries to predict, if it its real or fake. The main idea is to train the generator so that it tries to maximize the probability of the discriminator making a mistake and thereby produces more and more realistic looking generated samples from the source domain. The GANs are formulated as a minmax game in which the generator tries to maximize and the discriminator tries to minimize the following loss equation:

$$\mathcal{L}_{\text{GAN}}(G, D) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (2)$$

Here, G is the generator, D is the discriminator, x represents real data samples, and z denotes random noise. The minmax loss is a combination of two Binary Cross-Entropy [BCE] losses: the first term corresponds to the BCE loss for the real data [$\text{BCE}(D(x), 1)$] while the second term corresponds to the BCE loss for the generated data [$\text{BCE}(D(G(z)), 0)$]. This formulation encourages the generator to produce data that resembles the real distribution [$\log(1-D(G(z)))$] while simultaneously urging the discriminator to classify real and generated data accurately [$\log D(x)$].

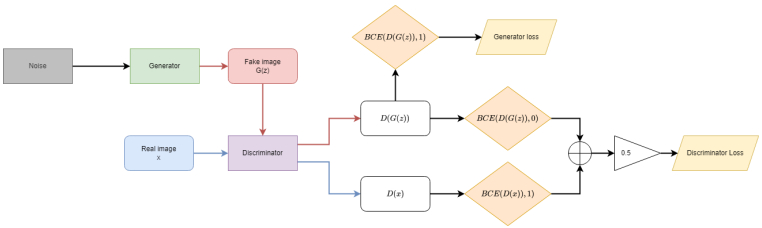


Figure 9: GAN architecture flowchart

In the implementation the BCE losses are used to train the generator and the discriminator as seen in the graphic above. One of the main advantages of

this adversarial loss of the GAN is that the network can be trained on unpaired data samples. When a GAN is used to generate an image from the target domain based on a specific instance from the source domain the traditional the architecture is called a conditional GAN, one popular implementation is the pix2pix-network.

4.4.2 Pix2Pix

The Pix2Pix network [5] is a type of conditional GAN 4.4.1 designed specifically for image-to-image translation tasks. Instead of generating random samples, the Pix2Pix network takes an input image and maps it to a corresponding output image in a structured way.

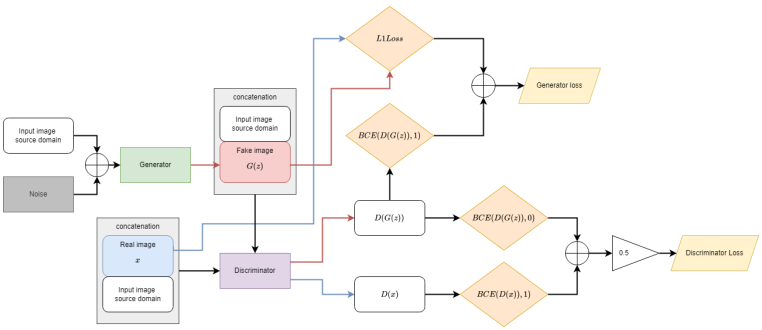


Figure 10: Pix2pix architecture flowchart

So, the Pix2Pix network is extending the GAN framework shown in 4.4.2. To noise input of the vanilla GAN, the image of the source domain of the image pair is added which is then used as the input for the generator. The discriminator functions similar to the traditional GAN and tries to distinguish between real and generated images. But here the generated and the real image of the target domain are concatenated with the corresponding image of the source domain. This joint representation is than used as the input of the discriminator. Furthermore, the Pix2Pix network supplements the adversarial loss of the generator with mean absolute error (L1-loss) between the generated image and the real image of the target domain.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{3}$$

This loss helps ensure that the generated images are structurally aligned with the ground truth images.
As generator, the U-Net architecture 4.1 is chosen and an additional discriminator was implemented.

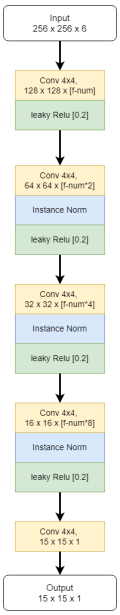


Figure 11: Discriminator flowchart

After an initial convolution with leaky ReLu the discriminator has three single convolution blocks with instance norm and leaky ReLu layer which reduce the height and width of the image by half and double the feature dimension. The last layer will reduce the number of channels to one so that an [15 x 15 x 1] image forms the output of the discriminator.

4.5 Diffusion Model

The concept of diffusion as a technique for generative models was introduced by Sohl-Dickstein and his colleagues at the International conference on machine learning in 2015 [11] [7]. The idea is to leverage the concept of probabilistic diffusion to learn mappings between the source domain and the target domain. In the forward process noise is added iteratively to the original image x_0. In this process the image with the added noise is calculated by the product of it's predecessors.

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{4}$$

The noise is sampled depending on the previous image x_{t-1} and a variance schedule β_t.

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{5}$$

The noise has a Gaussian distribution where with the mean (μ) and the variance (σ) defined as follows:

$$\mu = \sqrt{1-\beta_t}x_{t-1} \tag{6}$$

$$\sigma = \beta_t I \tag{7}$$

To train the diffusion model the number of steps T was set to 1000. Although diffusion models are often used for text to image generation to adapt it to image translation the image from the source domain is used as the condition instead of the text. The image from the source domain is added to the noise at X_T. One draw back of the Diffusion Model is that the sampling process takes more time than in the other presented models due to the huge number of sampling steps.

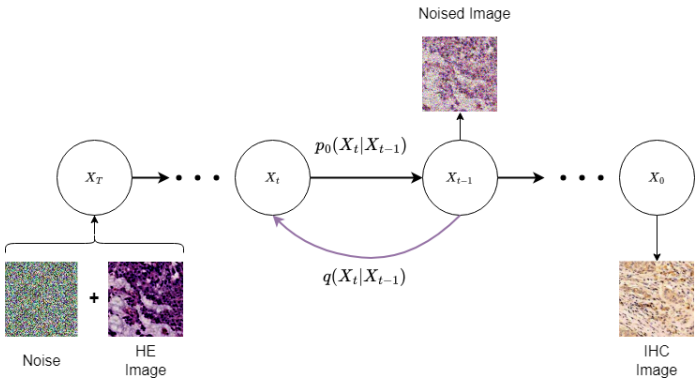


Figure 12: Diffusion Model flow chart

5 Data Set

The data set of the BCI-challenge [8] provided a train set (3396), test set (977), and validation set (500). The original images size is full HDD [1024 x 1024]. To keep the model sizes small the images were divided into 16 unique [256 x 256] patches which were used as the input of the networks. Due to the fact that in the test set no IHC stained images were released, the testing is completely done on the validation set. The last three patches of each training image are used as validation set. Overall this yields 62336 image pairs of the size [256 x 256]. These divide into 54336 training pairs and 8000 testing pairs. The scores for the the IHC stained images are distributed as follows for train,val and test set:

score	train set	test set
0 (negative)	2752	480
1+ (weak)	12864	1840
2+ (moderate)	23616	3536
3+ (strong)	15104	2144

Table 1: data set composition

6 Experiment Design

In this chapter the design and the execution of the experiments are presented. The experiments will compare the performance of the approaches described in the previous chapter **??**.
Each experiment will run for 100 epochs while the epochs represent a cycle of all images in the train set (3396). To enlarge the number of unique images each image of the size[1024,1024,3] is divided into 16 patches instead of resizing the image. For each new epoch the next image patch is chosen so that the same patch of the same image will return as input for the network after 16 epochs. For image augmentation the loader will randomly flip both HE and IHC image patch vertically and horizontally. Additionally, some colorjitter with brightness[0.8/1.2], contrast[1/2] and saturation[0.8/1.2] is applied. After each epoch the mean MSE, SSIM and PSNR are stored. Every 5 epochs the results of the network on the entire test set (all patches) are stored and a checkpoint is saved. The results of the networks will be compared on the separate test set while varying the architecture and some key features within.
Each experiment will run for 100 epochs and the network with the lowest error score 8 on the validation set will be used as the best result for that approach. The error score is calculated as follows:

$$errorscore = MSE + (1 - SSIM) \tag{8}$$

To evaluate the the networks the MSE, SSIM and PSNR is calculated for all images and for each group of images with the same IHC score on all patches of the train set, validation-set and test set. Plots for the MSE, SSIM and PSNR of the network during the training are saved.

6.1 U-Net Experiments

The U-Nets which have been investigated all share the same encoder and decoder blocks **??** with the following parameters:

parameter	convlayer	maxpool layer	transpose convlayer
kernel size	3	2	2
padding	1	0	0
stride	1	1	2

Table 2: U-net block parameters

As well all U-nets use the same Adam optimiser which was chosen for all generative networks with a learn rate = 0.0002 and a $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In the different U-Net implementations the number of steps and the number of features were varied to determine the complexity necessary for the image translation task.

architecture	number of steps	number of feature
U-Net S	3	16
U-Net M	4	32
U-Net L	5	64

Table 3: U-net architecture parameters

6.2 Visual Transformer Experiments

Visual transformer architecture experiments pretests showed that the patch size is crucial for the image translation task. The network has difficulty to show details in the generated image that lay within one patch. Therefore the patch size was chosen as small as possible for the image size and the hardware which was used. The patch size for all ViT architectures is [4,4]. Furthermore, the dropout rate in the attention layer was set to 0.1, the MLP ratio to 4 and the dropout rate to 0.2.
In the different visual transformer implementations the number of transformer blocks and the number of heads were varied to determine the complexity necessary for the image translation task.

architecture	number of ViT blocks	number of heads
ViT S	1	2
ViT M	2	4

Table 4: ViT architecture parameters

In addition to the ViT implementations the a Swin Transformer was implemented with two stages and a hidden dimension of 32.

6.3 Pix2Pix Experiments

The pix2pix architecture was trained with 3 different types of generators: The U-Net with 5 steps and a hidden dimension of 64, Vision Transformer with a patchsize of 4 and 1 transformer blocks 4.2 and the Swin Transformer with 2 stages. The Discriminator will remain the same for all the experiments built from 4 convolution blocks. For the U-Net and the Transformer generators the architectures where chosen which performed the best individually without the GAN structure.

6.4 Diffusion Model experiments

The Diffusion Model model was implemented with 1000 denoising steps. The HE-stained image was added as condition to the noise 4.5. The model to perform the denoising is a U-Net as described in 9.1.1 with three down sampling steps and three up sampling steps and an additional self attention layer for each step. The sampling time of the Diffusion Model is considerably longer for the other models, this is why a loss graph in the appendix is not provided.

6.5 Evaluation

The evaluation of the experiments is constructed twofold, encompassing the quantitative metrics of the generated images and a qualitative evaluation of the images. In the first part, the evaluation of all the architectures described in the previous chapter are evaluated with the following three metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM).
In the second part of the evaluation, the generated images are assessed qualitatively through visual comparison to the original IHC image. Additionally, a classifier was trained on the original IHC images to predict the IHC score and than was used to predict the score of the generated images.

6.5.1 Quantitative Evaluation

The MSE, PSNR and SSIM are applied to all patches of the train and test set. To gain more insight on the strength and weaknesses of each architecture the metrics were also calculated for each subgroup with the same IHC-score.

The **Mean Squared Error** (MSE) is given by:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2 \tag{9}$$

where X_i and Y_i are the pixel values of the two images being compared, and N is the total number of pixels.

The **Peak Signal-to-Noise Ratio** (PSNR) is calculated as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \tag{10}$$

where MAX is the maximum possible pixel value.

The **Structural Similarity Index** (SSIM) is given by:

$$\text{SSIM}(X, Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{11}$$

where μ_X and μ_Y are the average pixel values, σ_X and σ_Y are the standard deviations, σ_{XY} is the covariance and c_1 and c_2 are constants to stabilize the division.
The plots show the development of these values during the training and a further analysis shows the performance of the net during the different epochs.

6.5.2 Qualitative Evaluation

The qualitative evaluation is split into two parts: first the output of each network is shown next to the HE-stained image and the ground truth. A sample image from each IHC-score is used to evaluate the network's performance for each score group. This allows a visual comparison between the ground truth and the image produced by the network.
Second: in an effort to evaluate the interpretive quality of the generated images a Classifier was trained to predict the IHC-score on the ground truth images. This Classifier is than used to predict the IHC-score of the generated images. The assumption being, that the better the classifier, on the generated, images the better is the quality of those images.
The classifier is based on previous work [6] by Saidul Kabir, Semir Vranic and their colleagues who investigated different deep learning approaches to classify IHC-stained images of breast tissue. Starting from the results of the paper, a Visual Transformer architecture was chosen. In addition to the weights pretrained on ImageNet [2], the network was fine tuned for 20 epochs on the train images resized to [256,256,3]. In the qualitative evaluation, the decision was made to use resized images rather than patches because the information of the IHC-score might not translate to each image patch. To assess the performance of the classifier, a confusion matrix for the scores is created:

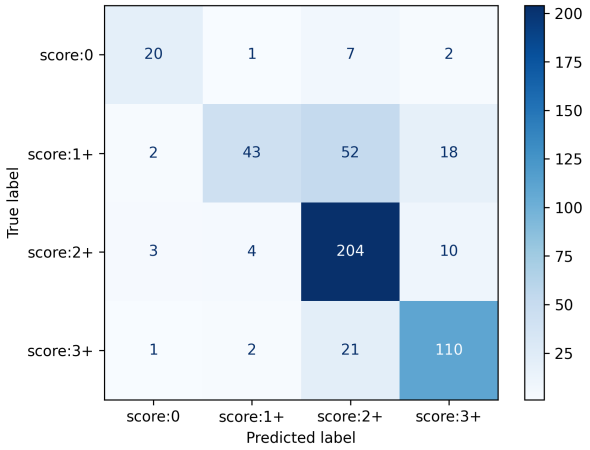


Figure 13: Confusion matrix for the classifier on the original IHC images

In addition the precision, the recall, the F1-score, and the support for each IHC-score is calculated:

IHC-score	precision	recall	f1-score	support+
score: 0	0.769	0.666	0.714	30
score: 1+	0.86	0.373	0.521	115
score: 2+	0.718	0.923	0.807	221
score: 3+	0.785	0.82	0.802	134

Table 5: classification report on original IHC images

The overall accuracy of the classifier on the original IHC-stained images is 75.4%.

7 Results

In the following chapter the results of the experiments are presented. In the first section the results of the quantitative evaluation are shown. Here Box-Whisker-Plots for each model architecture for the MSE, SSIM and PSNR performance are shown for each IHC-score group. In the later section all model architectures are compared based on the performance of the whole train set and test set. For The ease of comparing the results the performance values are also provided in table form. The loss graph of each training is shown in the appendix 9.
In the second section the qualitative results are shown. Here a sample of four generated images, one from each IHC-score, is shown for every network. Additionally the precision, recall, f1-score and support for the performance of the classifier on all four IHC-scoring groups is listed. The full confusion matrix for the performance of the classifier on the generated images for every network is shown in the appendix 9.

7.1 Quantitative Results

7.1.1 U-Net S Results

The small U-Net which was implemented with three steps and 16 features.
The average sampling time on test was 0.00386 seconds.
The network was training for 15.6932 hours.

MSE Evaluation:

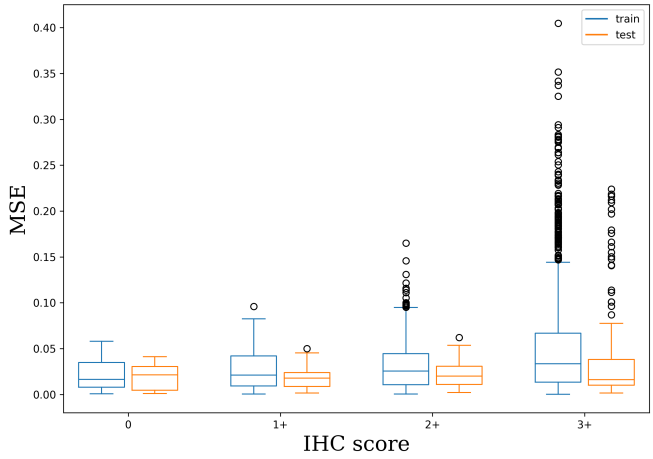


Figure 14: MSE performance of U-Net S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0258	0.018	0.0182	0.021	0.0419
	train	0.0357	0.0221	0.0256	0.0292	0.057
MSE_{var}	test	0.0011	0.0002	0.0001	0.0002	0.0032
	train	0.0017	0.0003	0.0003	0.0005	0.0043
MSE_{min}	test	0.0011	0.0011	0.0016	0.0022	0.0016
	train	0.0004	0.0007	0.0006	0.0004	0.0004
MSE_{max}	test	0.2238	0.0414	0.0499	0.0621	0.2238
	train	0.4048	0.0581	0.0959	0.1651	0.4048

Table 6: U-Net S MSE performance

SSIM Evaluation:

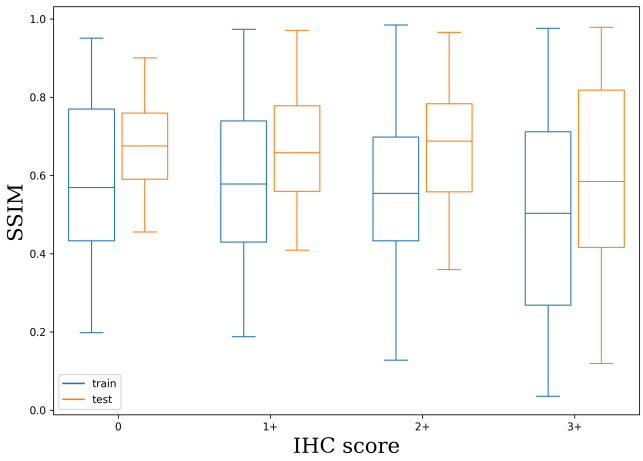


Figure 15: SSIM performance of U-Net S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.655	0.6754	0.6782	0.6756	0.5966
	train	0.556	0.5962	0.5869	0.5704	0.4999
$SSIM_{var}$	test	0.0309	0.0131	0.0195	0.0226	0.0537
	train	0.045	0.0425	0.0359	0.0342	0.0654
$SSIM_{min}$	test	0.1193	0.4551	0.4089	0.3592	0.1193
	train	0.0356	0.198	0.1875	0.1276	0.0356
$SSIM_{max}$	test	0.9789	0.9004	0.9711	0.9657	0.9789
	train	0.9842	0.9507	0.9734	0.9842	0.9763

Table 7: U-Net S SSIM performance

PSNR Evaluation:

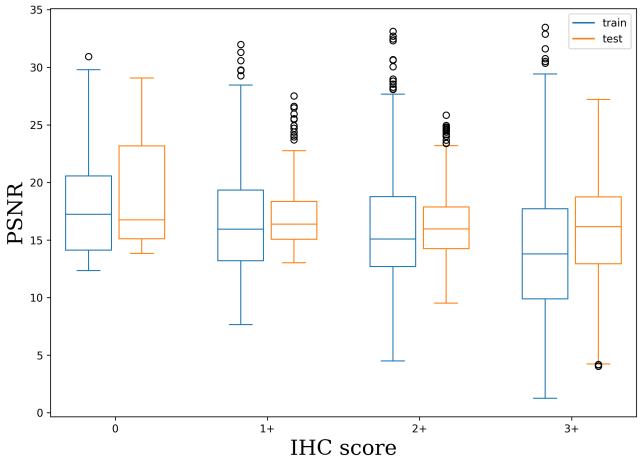


Figure 16: PSNR performance of U-Net S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	16.5985	19.1966	17.4537	16.5401	15.379
	train	15.5936	17.758	16.4481	15.7605	14.2106
$PSNR_{var}$	test	18.0687	20.538	12.7051	11.2911	29.6655
	train	24.215	16.6868	18.3067	20.5543	32.9108
$PSNR_{min}$	test	4.0467	13.8352	13.0181	9.5103	4.0467
	train	1.2686	12.3556	7.6645	4.5033	1.2686
$PSNR_{max}$	test	29.0639	29.0639	27.5233	25.8535	27.2005
	train	33.4873	30.9356	31.9827	33.1288	33.4873

Table 8: U-Net S PSNR performance

7.1.2 U-Net M Results

The medium U-Net which was implemented with four steps and 32 features.
The average sampling time on test was 0.00554 seconds.
The network was training for 16.01075 hours.

MSE Evaluation:

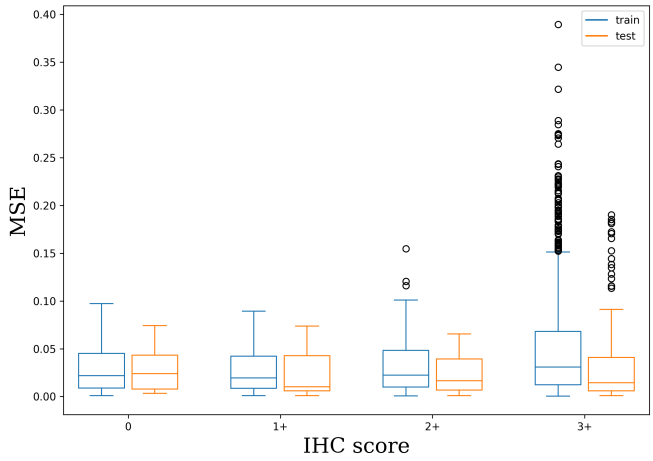


Figure 17: MSE performance of U-Net M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0267	0.0274	0.0226	0.0237	0.0351
	train	0.0358	0.0295	0.0276	0.0305	0.0522
MSE_{var}	test	0.0009	0.0004	0.0004	0.0004	0.0023
	train	0.0015	0.0006	0.0006	0.0006	0.0034
MSE_{min}	test	0.001	0.0032	0.001	0.001	0.001
	train	0.0005	0.0009	0.001	0.0006	0.0005
MSE_{max}	test	0.1901	0.0741	0.0738	0.0655	0.1901
	train	0.3896	0.0973	0.0894	0.1547	0.3896

Table 9: U-Net M MSE performance

SSIM Evaluation:

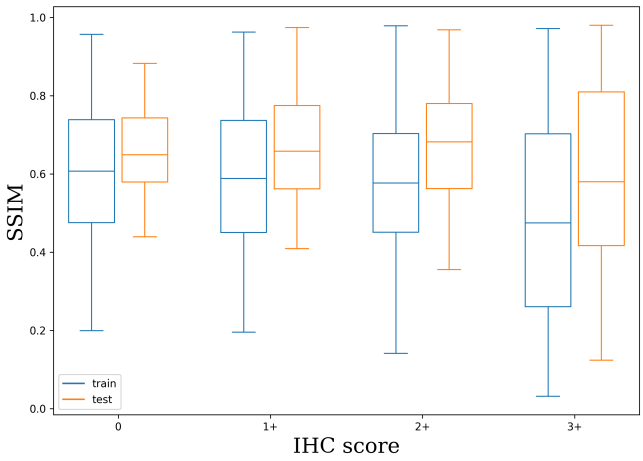


Figure 18: SSIM performance of U-Net M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.6528	0.6644	0.6765	0.674	0.5951
	train	0.5599	0.6086	0.5941	0.5815	0.4882
$SSIM_{var}$	test	0.0297	0.0117	0.0186	0.0219	0.0515
	train	0.0435	0.033	0.032	0.0321	0.0656
$SSIM_{min}$	test	0.1235	0.4389	0.4088	0.3555	0.1235
	train	0.0317	0.1989	0.1955	0.1413	0.0317
$SSIM_{max}$	test	0.9801	0.8823	0.974	0.9679	0.9801
	train	0.9787	0.9568	0.9624	0.9787	0.9714

Table 10: U-Net M SSIM performance

PSNR Evaluation:

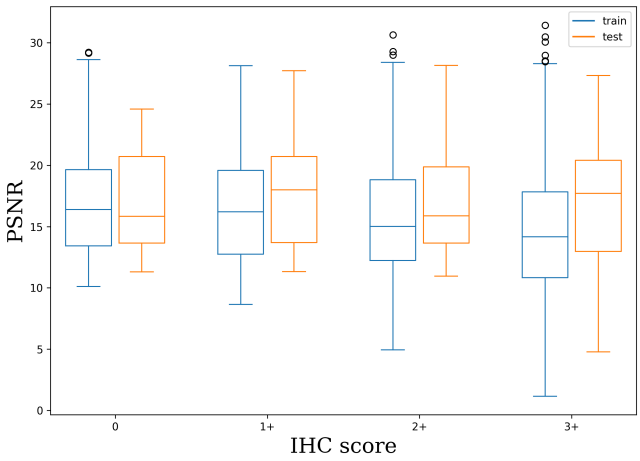


Figure 19: PSNR performance of U-Net M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	17.0685	16.9745	17.6991	17.0656	16.5531
	train	15.6124	16.6467	16.51	15.6949	14.5305
$PSNR_{var}$	test	20.7141	14.151	17.0774	16.8987	30.9881
	train	20.9955	16.3504	17.6568	17.6658	27.8294
$PSNR_{min}$	test	4.7821	11.2998	11.3186	10.9587	4.7821
	train	1.165	10.1198	8.6415	4.9339	1.165
$PSNR_{max}$	test	28.1547	24.5812	27.7185	28.1547	27.331
	train	31.4364	29.2229	28.1223	30.6489	31.4364

Table 11: U-Net M PSNR performance

7.1.3 U-Net L Results

The largest U-net which was implemented with five steps and 64 features.
The average sampling time on test was 0.0079 seconds.
The network was training for 19.6704 hours.

MSE Evaluation:

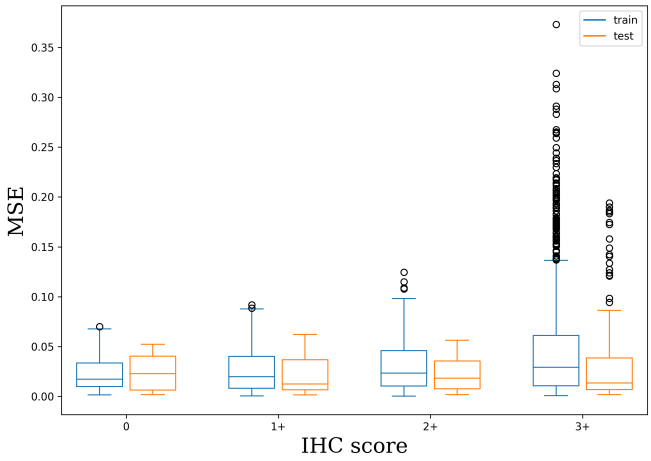


Figure 20: MSE performance of U-Net L on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0255	0.023	0.0206	0.022	0.0361
	train	0.0341	0.0239	0.0263	0.029	0.0506
MSE_{var}	test	0.0009	0.0003	0.0003	0.0003	0.0024
	train	0.0014	0.0003	0.0005	0.0005	0.0035
MSE_{min}	test	0.0016	0.0017	0.0016	0.0018	0.0019
	train	0.0004	0.0016	0.0006	0.0004	0.0007
MSE_{max}	test	0.194	0.0523	0.0622	0.0563	0.194
	train	0.3731	0.0701	0.0918	0.1245	0.3731

Table 12: U-Net L MSE performance

SSIM Evaluation:

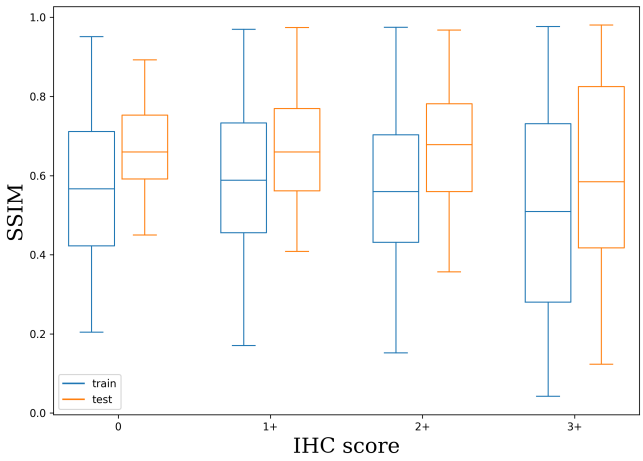


Figure 21: SSIM performance of U-Net L on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.656	0.6671	0.6791	0.6762	0.6004
	train	0.5593	0.5734	0.5925	0.5725	0.5076
$SSIM_{var}$	test	0.0305	0.0114	0.0193	0.0224	0.0536
	train	0.0442	0.0384	0.0318	0.0341	0.0676
$SSIM_{min}$	test	0.1236	0.4498	0.4087	0.3564	0.1236
	train	0.0427	0.2044	0.1707	0.1521	0.0427
$SSIM_{max}$	test	0.9801	0.8925	0.9739	0.9676	0.9801
	train	0.9766	0.9513	0.9698	0.9748	0.9766

Table 13: U-Net L SSIM performance

PSNR Evaluation:

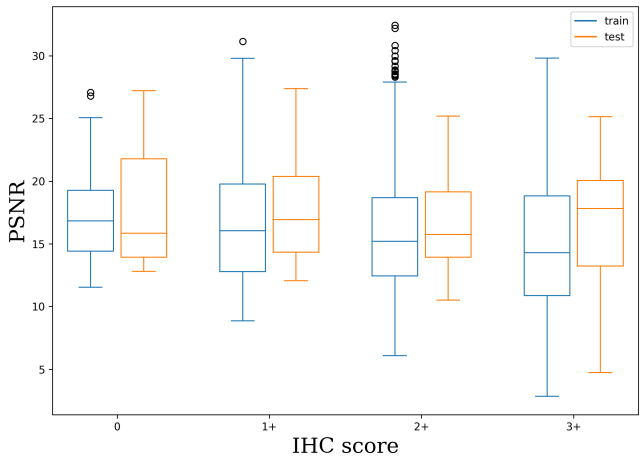


Figure 22: PSNR performance of U-Net L on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	16.8496	18.0544	17.4953	16.784	16.1339
	train	15.8534	17.181	16.5897	15.9108	14.8946
$PSNR_{var}$	test	16.814	18.6355	12.8498	11.3227	27.6627
	train	22.6271	12.3039	18.9154	18.9419	31.7241
$PSNR_{min}$	test	4.7249	12.8128	12.062	10.5166	4.7249
	train	2.848	11.5448	8.8506	6.0724	2.848
$PSNR_{max}$	test	27.3827	27.2116	27.3827	25.1885	25.1385
	train	32.4452	27.0778	31.1419	32.4452	29.8164

Table 14: U-Net L PSNR performance

7.1.4 ViT S Results

The slimmest Vision Transformer which was implemented with one block and two heads.
The average sampling time on test was 0.0023 seconds.
The network was training for 16.7576 hours.

MSE Evaluation:

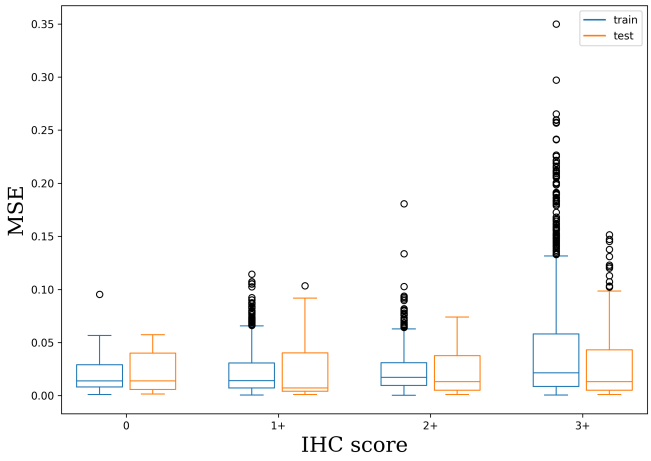


Figure 23: MSE performance of ViT S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0246	0.0231	0.0245	0.0215	0.0301
	train	0.0285	0.0194	0.0222	0.022	0.0456
MSE_{var}	test	0.0008	0.0003	0.0008	0.0004	0.0014
	train	0.0012	0.0002	0.0004	0.0003	0.0032
MSE_{min}	test	0.001	0.0015	0.001	0.0011	0.0011
	train	0.0005	0.0011	0.0005	0.0005	0.0005
MSE_{max}	test	0.1515	0.0574	0.1034	0.074	0.1515
	train	0.3498	0.0955	0.1143	0.1806	0.3498

Table 15: ViT S MSE performance

SSIM Evaluation:

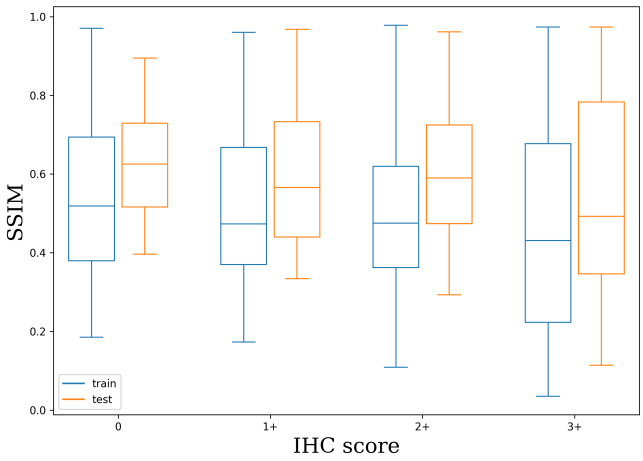


Figure 24: SSIM performance of ViT S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.5912	0.6288	0.5984	0.6072	0.5502
	train	0.4987	0.5474	0.5218	0.5049	0.4604
$SSIM_{var}$	test	0.0351	0.018	0.0276	0.0275	0.0555
	train	0.046	0.0419	0.0366	0.0363	0.0675
$SSIM_{min}$	test	0.1138	0.3963	0.3343	0.2926	0.1138
	train	0.0353	0.1849	0.1728	0.1089	0.0353
$SSIM_{max}$	test	0.9738	0.895	0.9677	0.9617	0.9738
	train	0.9785	0.9704	0.9602	0.9785	0.9736

Table 16: ViT S SSIM performance

PSNR Evaluation:

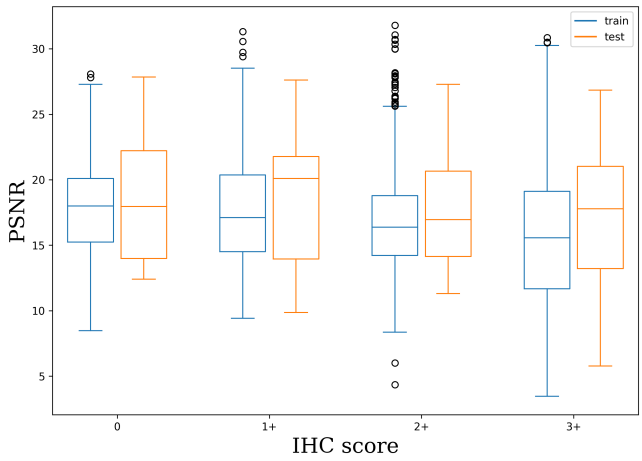


Figure 25: PSNR performance of ViT S on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	17.7149	18.1894	18.4709	17.6756	17.0247
	train	16.6306	18.0057	17.4625	16.7221	15.5285
$PSNR_{var}$	test	22.1715	20.1396	23.8252	16.368	29.7589
	train	19.0808	11.5887	16.3878	12.9581	30.151
$PSNR_{min}$	test	5.7678	12.4093	9.8542	11.3083	5.7678
	train	3.4729	8.4732	9.4177	4.3589	3.4729
$PSNR_{max}$	test	27.8415	27.8415	27.6219	27.2905	26.8397
	train	31.8049	28.0678	31.316	31.8049	30.8551

Table 17: ViT S PSNR performance

7.1.5 ViT M Results

The slimmest Vision Transformer which was implemented with two blocks and four heads.
The average sampling time on test was 0.0030 seconds.
The network was training for 17.5575 hours.

MSE Evaluation:

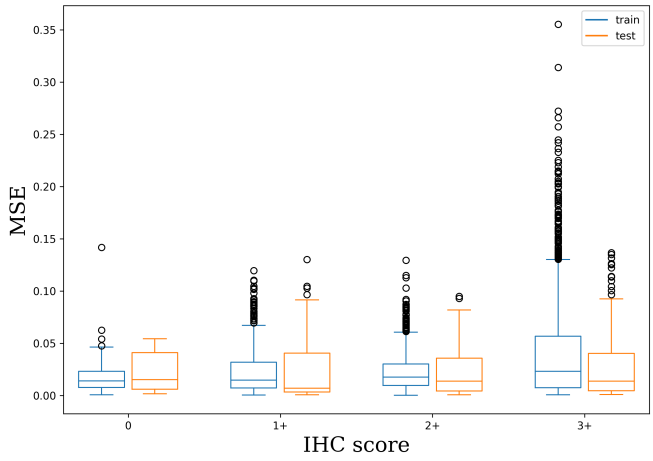


Figure 26: MSE performance of ViT M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0248	0.0233	0.0258	0.0216	0.0296
	train	0.0279	0.0178	0.0225	0.0218	0.044
MSE_{var}	test	0.0008	0.0004	0.001	0.0004	0.0013
	train	0.0011	0.0002	0.0004	0.0003	0.0028
MSE_{min}	test	0.0007	0.0017	0.0007	0.0008	0.0009
	train	0.0003	0.0007	0.0003	0.0003	0.0006
MSE_{max}	test	0.1366	0.0544	0.1301	0.095	0.1366
	train	0.3553	0.1417	0.1196	0.1295	0.3553

Table 18: ViT M MSE performance

SSIM Evaluation:

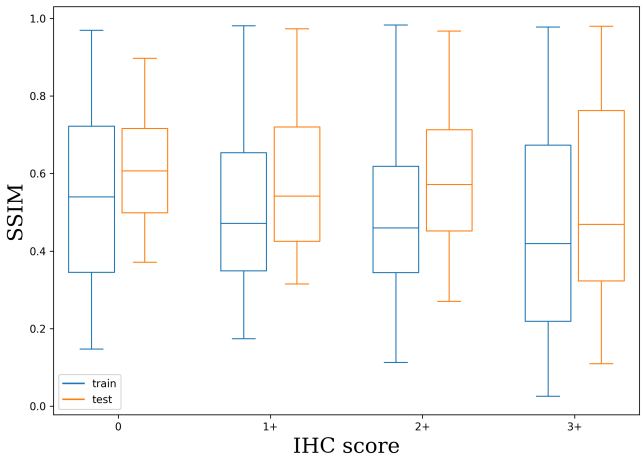


Figure 27: SSIM performance of ViT M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.5735	0.6161	0.5788	0.5882	0.5352
	train	0.4919	0.5509	0.5082	0.4993	0.4558
$SSIM_{var}$	test	0.0369	0.0199	0.0297	0.0297	0.0566
	train	0.0483	0.05	0.0378	0.0396	0.0684
$SSIM_{min}$	test	0.1097	0.3715	0.3151	0.2705	0.1097
	train	0.0258	0.1473	0.1737	0.1125	0.0258
$SSIM_{max}$	test	0.9796	0.8971	0.973	0.9677	0.9796
	train	0.9828	0.9696	0.981	0.9828	0.978

Table 19: ViT M SSIM performance

PSNR Evaluation:

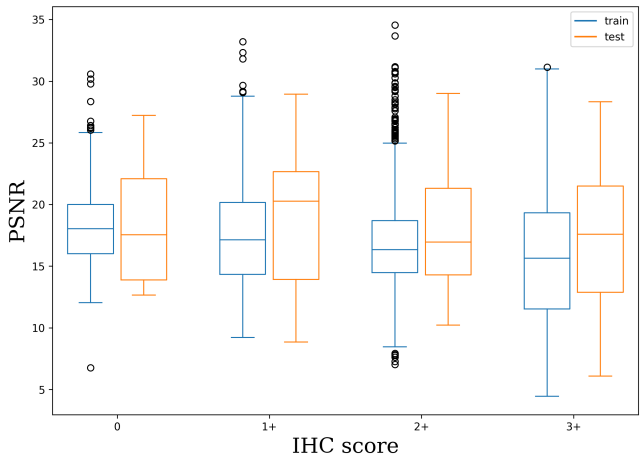


Figure 28: PSNR performance of ViT M on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	17.9049	18.0898	18.6824	17.8889	17.2225
	train	16.7476	18.4879	17.441	16.8098	15.7428
$PSNR_{var}$	test	24.8561	19.3982	27.6756	19.0727	32.2041
	train	19.7449	14.31	16.6188	13.7941	30.7251
$PSNR_{min}$	test	6.0902	12.646	8.8579	10.2236	6.0902
	train	4.4588	6.7591	9.2222	7.0259	4.4588
$PSNR_{max}$	test	28.9994	27.233	28.938	28.9994	28.3427
	train	34.5451	30.5871	33.2085	34.5451	31.129

Table 20: ViT M PSNR performance

7.1.6 Swin Transformer Results

The Swin Transformer was implemented with two stages and 32 featuresin the hidden dimension.
The average sampling time on test was 0.01749 seconds.
The network was training for 15.5623 hours.

MSE Evaluation:

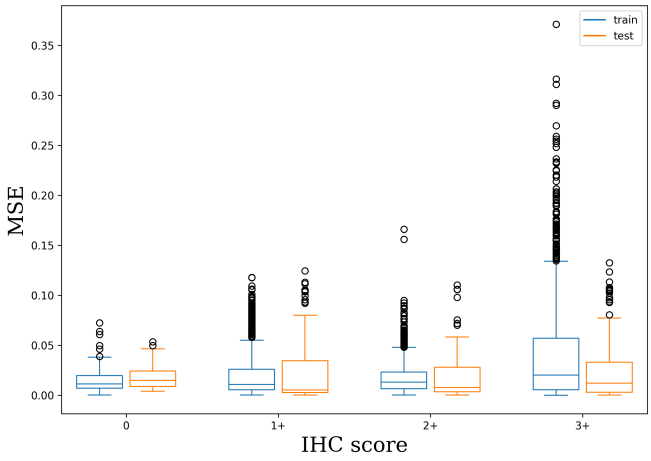


Figure 29: MSE performance of Swin T on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0213	0.0188	0.0245	0.0171	0.0262
	train	0.0254	0.0146	0.0214	0.0173	0.0432
MSE_{var}	test	0.0007	0.0002	0.0011	0.0004	0.0011
	train	0.0013	0.0001	0.0006	0.0002	0.0032
MSE_{min}	test	0.0001	0.0039	0.0001	0.0002	0.0001
	train	0.0001	0.0002	0.0001	0.0001	0.0001
MSE_{max}	test	0.1324	0.0534	0.1245	0.1103	0.1324
	train	0.3713	0.0724	0.1178	0.1661	0.3713

Table 21: Swin T MSE performance

SSIM Evaluation:

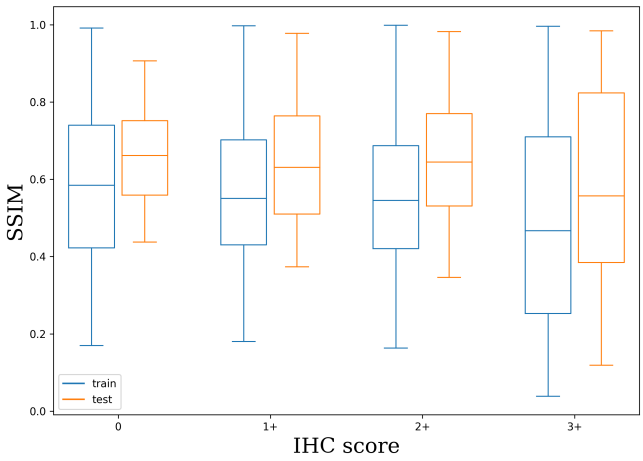


Figure 30: SSIM performance of Swin T on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.635	0.6573	0.647	0.6553	0.5864
	train	0.5466	0.5877	0.576	0.5622	0.4896
$SSIM_{var}$	test	0.033	0.0151	0.0241	0.0252	0.0543
	train	0.0467	0.0404	0.0335	0.0365	0.0704
$SSIM_{min}$	test	0.1186	0.4375	0.3735	0.3462	0.1186
	train	0.0387	0.1698	0.1803	0.1631	0.0387
$SSIM_{max}$	test	0.9843	0.9064	0.9779	0.9824	0.9843
	train	0.9988	0.9914	0.9974	0.9988	0.9956

Table 22: Swin T SSIM performance

PSNR Evaluation:

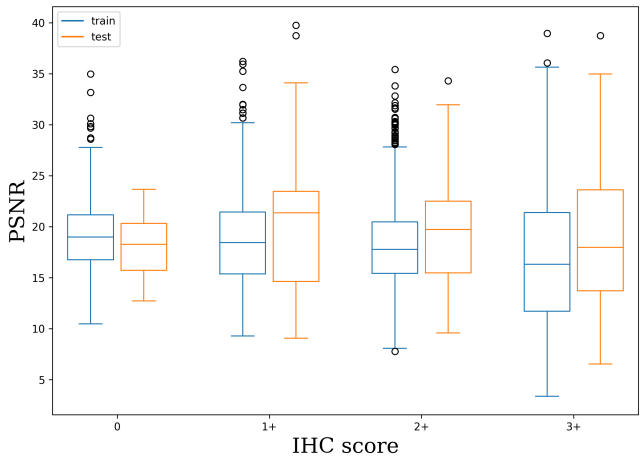


Figure 31: PSNR performance of Swin T on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	19.1873	18.1797	19.8222	19.4497	18.4351
	train	17.9043	19.3605	18.4484	18.1968	16.7182
$PSNR_{var}$	test	31.5865	9.5398	39.6603	23.0621	42.3997
	train	25.3096	16.6448	22.7284	15.3632	42.4596
$PSNR_{min}$	test	6.5267	12.7228	9.0491	9.5748	6.5267
	train	3.374	10.4797	9.2897	7.7631	3.374
$PSNR_{max}$	test	39.7708	23.6588	39.7708	34.3115	38.7342
	train	38.9582	34.9876	36.2268	35.4283	38.9582

Table 23: Swin T PSNR performance

7.1.7 Pix2Pix U-Net Results

The U-Net was implemented with five steps and 64 features.
The average sampling time on test was 0.0070 seconds.
The network was training for 19.8051 hours.

MSE Evaluation:

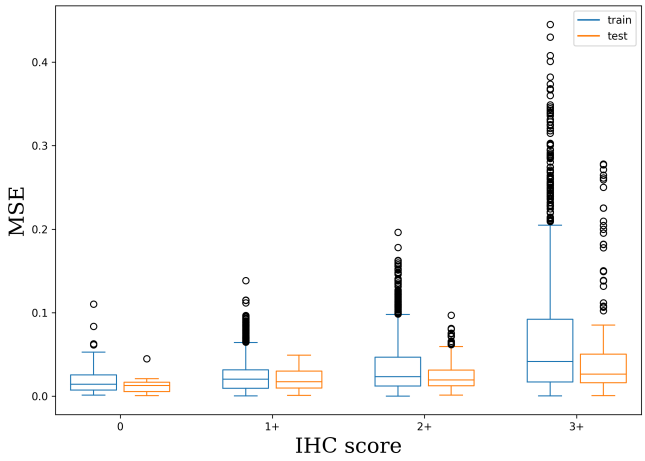


Figure 32: MSE performance of Pix2Pix U-Net on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0314	0.0119	0.0201	0.0248	0.0563
	train	0.0422	0.0182	0.0255	0.034	0.0737
MSE_{var}	test	0.0018	0.0001	0.0002	0.0003	0.0051
	train	0.0029	0.0002	0.0005	0.001	0.0072
MSE_{min}	test	0.0006	0.0006	0.0009	0.0012	0.0006
	train	0.0003	0.0013	0.0005	0.0003	0.0004
MSE_{max}	test	0.278	0.0448	0.0492	0.097	0.278
	train	0.4451	0.1101	0.1384	0.1961	0.4451

Table 24: Pix2Pix U-Net MSE performance

SSIM Evaluation:

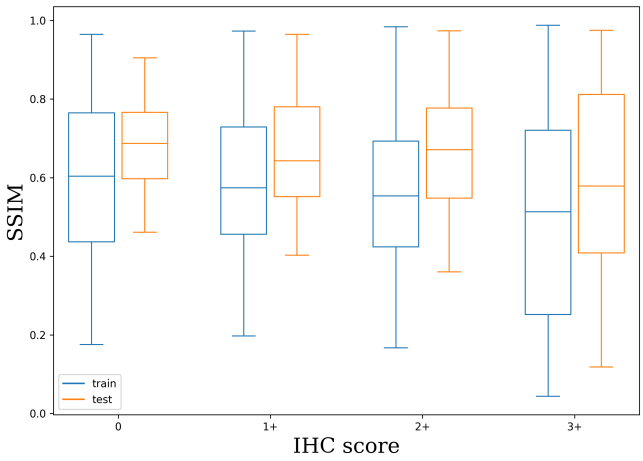


Figure 33: SSIM performance of Pix2Pix U-Net on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.6493	0.6834	0.6716	0.6694	0.5893
	train	0.5551	0.6034	0.5913	0.5656	0.4989
$SSIM_{var}$	test	0.0311	0.0129	0.0195	0.0228	0.0541
	train	0.0459	0.0418	0.0321	0.0349	0.0705
$SSIM_{min}$	test	0.118	0.4612	0.4026	0.36	0.118
	train	0.044	0.175	0.1971	0.1672	0.044
$SSIM_{max}$	test	0.9749	0.9051	0.9649	0.9739	0.9749
	train	0.9881	0.9644	0.9729	0.9837	0.9881

Table 25: Pix2Pix U-Net SSIM performance

PSNR Evaluation:

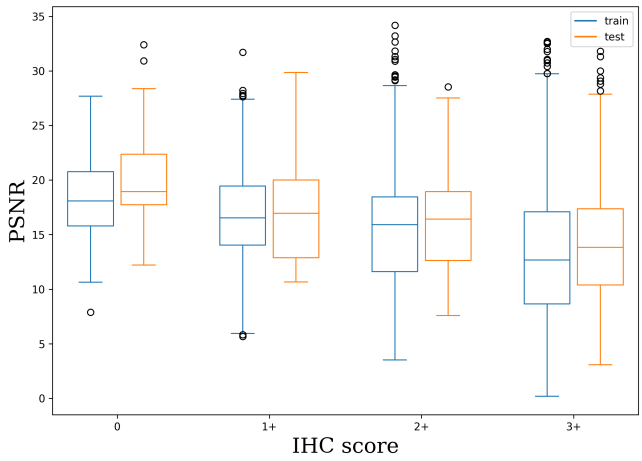


Figure 34: PSNR performance of Pix2Pix U-Net on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	16.0806	20.5978	17.0044	16.0713	14.292
	train	15.1924	18.2723	16.4092	15.4384	13.2102
$PSNR_{var}$	test	28.3965	19.1289	19.2463	20.2243	43.3022
	train	29.0535	13.8923	19.7882	26.3317	36.9496
$PSNR_{min}$	test	3.0624	12.2036	10.6598	7.5768	3.0624
	train	0.1934	7.8944	5.6583	3.5045	0.1934
$PSNR_{max}$	test	32.3978	32.3978	29.8557	28.541	31.8067
	train	34.2011	27.6941	31.709	34.2011	32.7113

Table 26: Pix2Pix U-Net PSNR performance

7.1.8 Pix2Pix ViT Results

The U-Net was implemented with one block and two heads.
The average sampling time on test was 0.0021 seconds.
The network was training for 20.8194 hours.

MSE Evaluation:

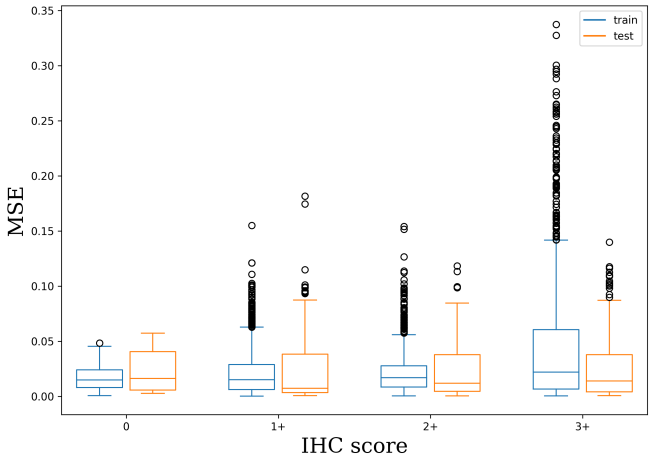


Figure 35: MSE performance of Pix2Pix ViT on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0259	0.0246	0.0292	0.0227	0.0286
	train	0.0287	0.0172	0.0225	0.021	0.0479
MSE_{var}	test	0.0009	0.0004	0.0015	0.0005	0.0011
	train	0.0015	0.0001	0.0005	0.0003	0.0039
MSE_{min}	test	0.0004	0.0026	0.0006	0.0004	0.0007
	train	0.0003	0.0006	0.0003	0.0003	0.0003
MSE_{max}	test	0.1815	0.0573	0.1815	0.1182	0.1398
	train	0.3374	0.0484	0.1549	0.154	0.3374

Table 27: Pix2Pix ViT MSE performance

SSIM Evaluation:

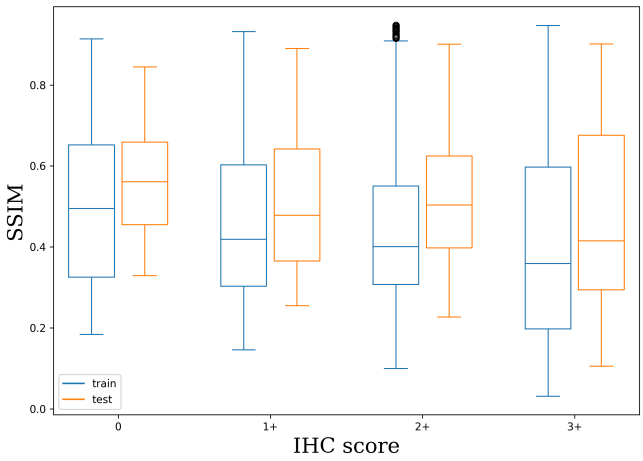


Figure 36: SSIM performance of Pix2Pix ViT on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.5122	0.5648	0.5134	0.5243	0.4794
	train	0.4402	0.5022	0.4579	0.445	0.4065
$SSIM_{var}$	test	0.0327	0.019	0.0276	0.0271	0.0475
	train	0.0421	0.039	0.035	0.035	0.0578
$SSIM_{min}$	test	0.105	0.3293	0.2551	0.2269	0.105
	train	0.0312	0.184	0.1457	0.0993	0.0312
$SSIM_{max}$	test	0.9017	0.8445	0.8902	0.9009	0.9017
	train	0.9479	0.9138	0.9318	0.9479	0.9467

Table 28: Pix2Pix ViT SSIM performance

PSNR Evaluation:

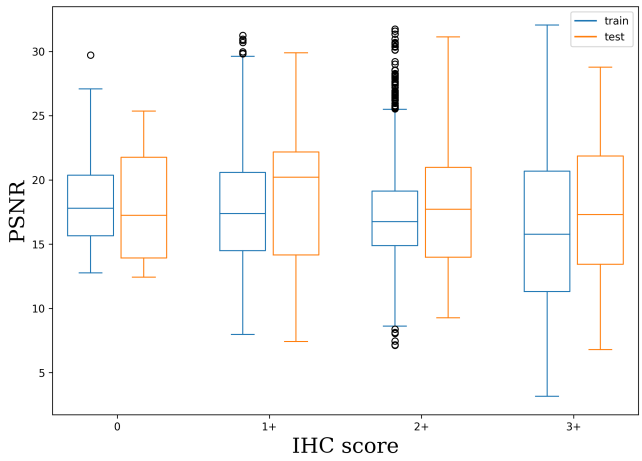


Figure 37: PSNR performance of Pix2Pix ViT on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	17.9587	17.8055	18.5427	18.021	17.3891
	train	17.0202	18.2584	17.6815	17.2119	15.9318
$PSNR_{var}$	test	27.0849	18.8107	30.7923	22.0596	33.4146
	train	22.0608	10.9306	18.5682	14.698	36.6819
$PSNR_{min}$	test	6.7917	12.4207	7.4113	9.2728	6.7917
	train	3.1627	12.7628	7.9611	7.1285	3.1627
$PSNR_{max}$	test	31.1446	25.3663	29.9032	31.1446	28.7754
	train	32.0561	29.7208	31.2515	31.7488	32.0561

Table 29: Pix2Pix ViT PSNR performance

7.1.9 Pix2Pix Swin Results

The U-Net was implemented with two stages and 32 features in the hidden dimension.
The average sampling time on test was 0.0339 seconds.
The network was training for 20.9548 hours.

MSE Evaluation:

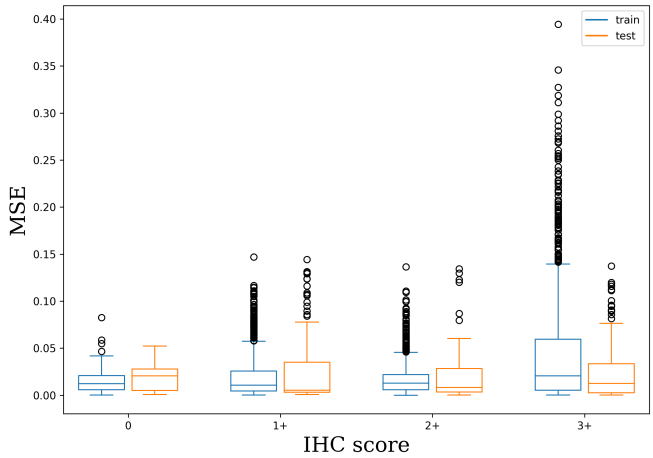


Figure 38: MSE performance of Pix2Pix Swin on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.0228	0.019	0.0278	0.0188	0.026
	train	0.0259	0.0143	0.0209	0.0168	0.0466
MSE_{var}	test	0.0009	0.0002	0.0015	0.0005	0.0011
	train	0.0016	0.0001	0.0006	0.0003	0.004
MSE_{min}	test	0.0001	0.0007	0.0007	0.0001	0.0002
	train	0.0	0.0001	0.0	0.0	0.0001
MSE_{max}	test	0.1443	0.0521	0.1443	0.1343	0.1372
	train	0.3945	0.0826	0.1471	0.1366	0.3945

Table 30: Pix2Pix Swin T MSE performance

SSIM Evaluation:

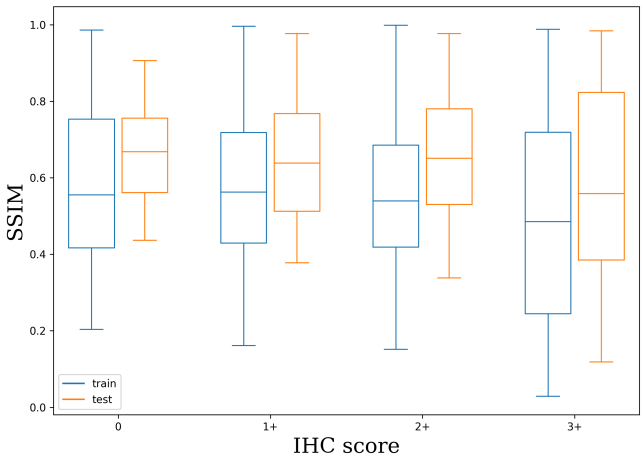


Figure 39: SSIM performance of Pix2Pix Swin on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.638	0.6608	0.6508	0.6584	0.5884
	train	0.5471	0.586	0.5784	0.5614	0.4911
$SSIM_{var}$	test	0.0335	0.0155	0.0246	0.0257	0.0547
	train	0.0471	0.0422	0.036	0.036	0.0703
$SSIM_{min}$	test	0.1182	0.4368	0.3775	0.3383	0.1182
	train	0.029	0.2037	0.1612	0.1516	0.029
$SSIM_{max}$	test	0.9843	0.9063	0.9772	0.9774	0.9843
	train	0.9992	0.9864	0.9965	0.9992	0.9886

Table 31: Pix2Pix Swin T SSIM performance

PSNR Evaluation:

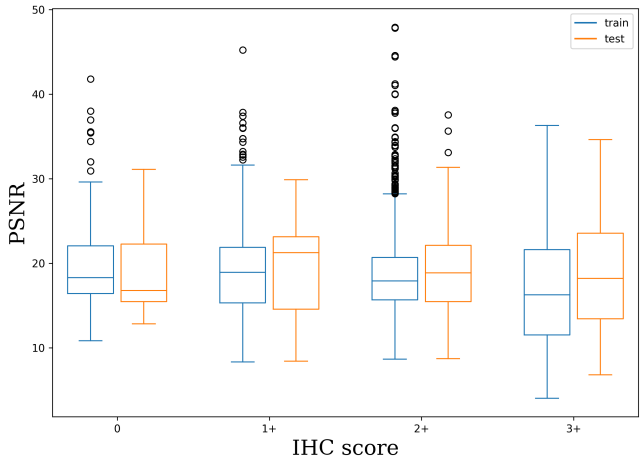


Figure 40: PSNR performance of Pix2Pix Swin on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	18.9865	19.1276	19.0947	19.1609	18.5743
	train	18.2683	19.8151	18.8874	18.6709	16.8298
$PSNR_{var}$	test	32.0458	23.985	32.58	25.0645	44.6713
	train	31.7939	25.8184	27.4298	23.3991	46.6401
$PSNR_{min}$	test	6.8059	12.828	8.407	8.7188	6.8059
	train	4.0391	10.8278	8.3248	8.6451	4.0391
$PSNR_{max}$	test	37.5522	31.1067	29.8947	37.5522	34.6277
	train	47.9374	41.8055	45.2284	47.9374	36.3059

Table 32: Pix2Pix Swin T PSNR performance

7.1.10 Diffusion Model Results

The Diffusion Model implemented with 1000 denoising steps.
The average sampling time on test was 71.58 seconds.
The network was training for 36.4 hours.
Due to the long sampling time, the evaluation of the Diffusion Model is more sparse. To evaluate the images in the train and test set, where resized to calculate the metrics instead of evaluating all patches.

MSE Evaluation:

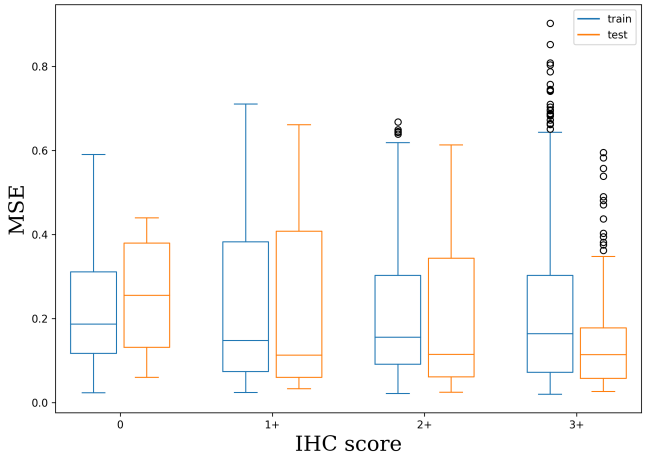


Figure 41: MSE performance of Diffusion on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE_{mean}	test	0.198	0.2526	0.2297	0.2021	0.152
	train	0.2143	0.2182	0.2322	0.2025	0.2167
MSE_{var}	test	0.0275	0.02	0.0421	0.0257	0.0162
	train	0.0263	0.0179	0.036	0.0191	0.0305
MSE_{min}	test	0.0246	0.0599	0.0333	0.0246	0.0264
	train	0.0203	0.0233	0.0243	0.0217	0.0203
MSE_{max}	test	0.6614	0.4393	0.6614	0.6131	0.595
	train	0.9028	0.5901	0.7101	0.668	0.9028

Table 33: Diffusion Model MSE performance

SSIM Evaluation:

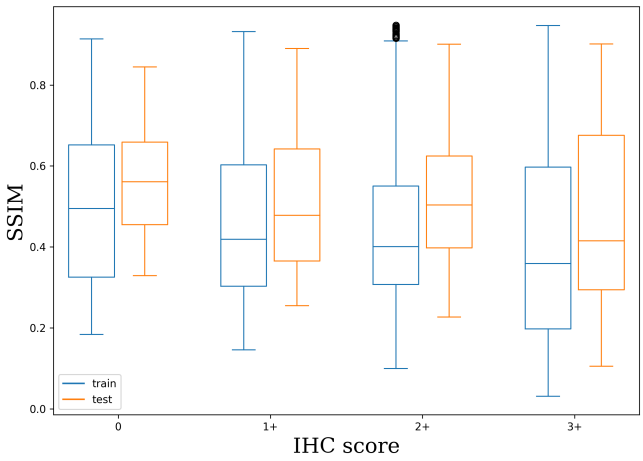


Figure 42: SSIM performance of Pix2Pix ViT on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$SSIM_{mean}$	test	0.0517	0.1214	0.0495	0.0511	0.0387
	train	0.0587	0.1132	0.0577	0.0592	0.0486
$SSIM_{var}$	test	0.0037	0.0078	0.0035	0.0031	0.0027
	train	0.0066	0.0109	0.0058	0.0074	0.0045
$SSIM_{min}$	test	-0.005	0.008	0.0025	0.0019	-0.005
	train	-0.0126	-0.0126	-0.0051	-0.0065	-0.0114
$SSIM_{max}$	test	0.3343	0.3343	0.2496	0.2498	0.2726
	train	0.4319	0.3888	0.4206	0.4319	0.3589

Table 34: Diffusion Model SSIM performance

PSNR Evaluation:

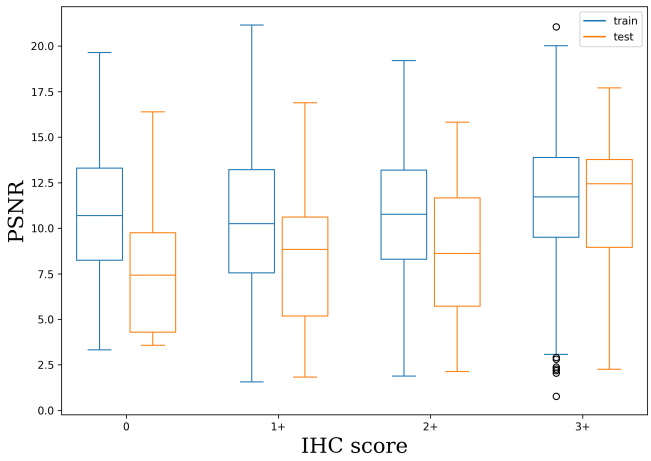


Figure 43: PSNR performance of diffusion on test set and train set

metric	set	total	score:0	score:1+	score:2+	score:3+
$PSNR_{mean}$	test	9.3159	7.9438	8.2956	8.6915	11.5286
	train	10.8418	10.6897	10.2971	10.6292	11.6658
$PSNR_{var}$	test	13.1822	15.082	12.0449	11.3606	9.8833
	train	12.9219	14.0213	15.8659	11.8417	10.8967
$PSNR_{min}$	test	1.8297	3.5723	1.8297	2.1291	2.2548
	train	0.7811	3.3284	1.5593	1.8759	0.7811
$PSNR_{max}$	test	17.7028	16.3902	16.8936	15.8248	17.7028
	train	21.1493	19.6476	21.1493	19.1979	21.0569

Table 35: Diffusion Model PSNR performance

7.1.11 Summary Quantitative Results

In this summery the performance of all networks in each metric. The Box-Whisker-Plot for each metric encompasses the full test set and train set data with all IHC-score. In the table below the results of all networks on the test set are shown with the best performance on all IHC-scores is marked for each metric.

MSE Evaluation Summary:

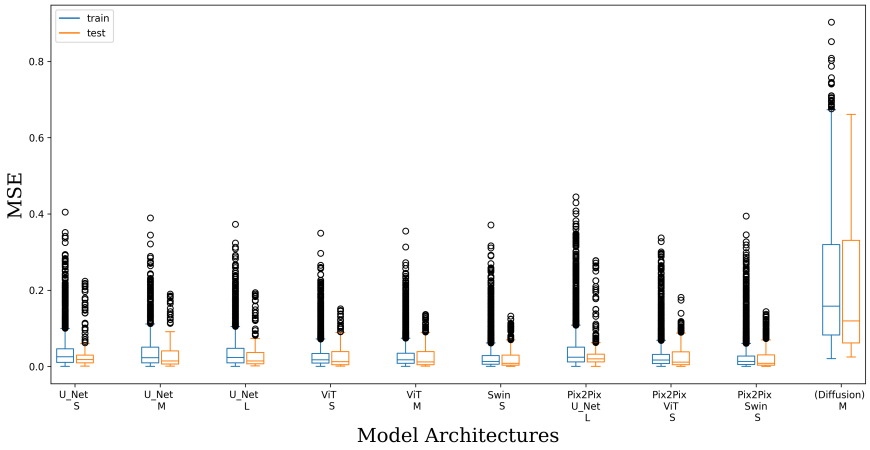


Figure 44: MSE performance of all networks on test set and train set

Network	total	score: 0	score: 1+	score: 2+	score: 3+
U-Net S	0.0258	0.018	0.0182	0.021	0.0419
U-Net M	0.0267	0.0274	0.0226	0.0237	0.0351
U-Net L	0.0255	0.023	0.0206	0.022	0.0361
ViT S	0.0246	0.0231	0.0245	0.0215	0.0301
ViT M	0.0248	0.0233	0.0258	0.0216	0.0296
Swin T	0.0213	0.0188	0.0245	0.0171	0.0262
Pix2Pix U-Net	0.0314	0.0119	0.0201	0.0248	0.0563
Pix2Pix ViT	0.0259	0.0246	0.0292	0.0227	0.0286
Pix2Pix Swin T	0.0228	0.019	0.0278	0.0188	0.026
(Diffusion Model)	0.198	0.2526	0.2297	0.2021	0.152

Table 36: MSE test performance all Networks

SSIM Evaluation Summary:

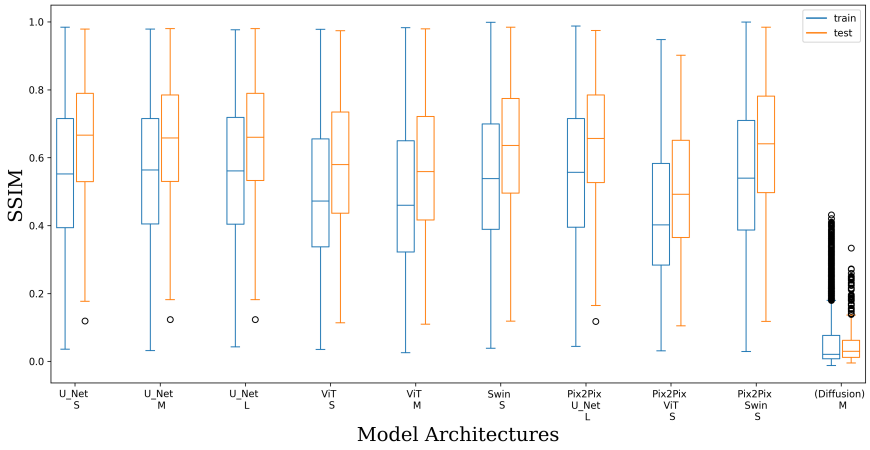


Figure 45: SSIM performance of all networks on test set and train set

Network	total	score: 0	score: 1+	score: 2+	score: 3+
U-Net S	0.655	0.6754	0.6782	0.6756	0.5966
U-Net M	0.6528	0.6644	0.6765	0.674	0.5951
U-Net L	0.656	0.6671	0.6791	0.6762	0.6004
ViT S	0.5912	0.6288	0.5984	0.6072	0.5502
ViT M	0.5735	0.6161	0.5788	0.5882	0.5352
Swin T	0.635	0.6573	0.647	0.6553	0.5864
Pix2Pix U-Net	0.6493	0.6834	0.6716	0.6694	0.5893
Pix2Pix ViT	0.5122	0.5648	0.5134	0.5243	0.4794
Pix2Pix Swin T	0.638	0.6608	0.6508	0.6584	0.5884
(Diffusion Model)	0.0517	0.1214	0.0495	0.0511	0.0387

Table 37: SSIM test performance all Networks

PSNR Evaluation Summary:

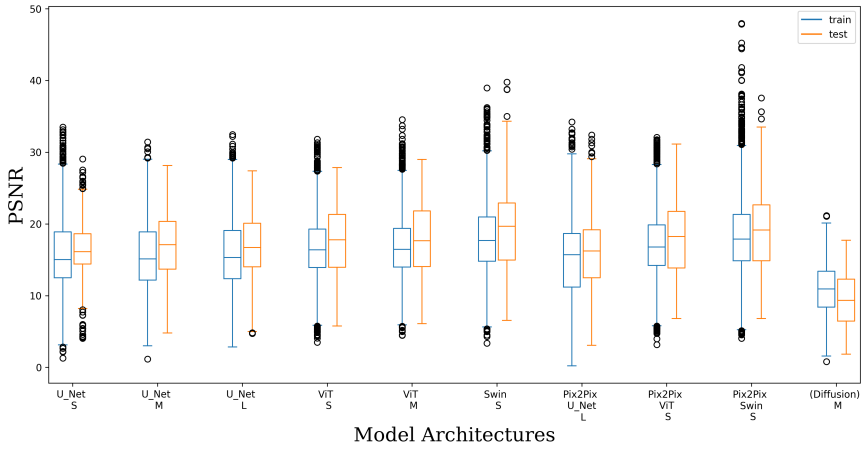


Figure 46: PSNR performance of all networks on test set and train set

Network	total	score: 0	score: 1+	score: 2+	score: 3+
U-Net S	16.5985	19.1966	17.4537	16.5401	15.379
U-Net M	17.0685	16.9745	17.6991	17.0656	16.5531
U-Net L	16.8496	18.0544	17.4953	16.784	16.1339
ViT S	17.7149	18.1894	18.4709	17.6756	17.0247
ViT M	17.9049	18.0898	18.6824	17.8889	17.2225
Swin T	19.1873	18.1797	19.8222	19.4497	18.4351
Pix2Pix U-Net	16.0806	20.5978	17.0044	16.0713	14.292
Pix2Pix ViT	17.9587	17.8055	18.5427	18.021	17.3891
Pix2Pix Swin T	18.9865	19.1276	19.0947	19.1609	18.5743
(Diffusion Model)	9.3159	7.9438	8.2956	8.6915	11.5286

Table 38: PSNR test performance all Networks

7.2 Qualitaive Results

7.2.1 U-Net S

The following plot is a sample of the images generated by the small U-Net.

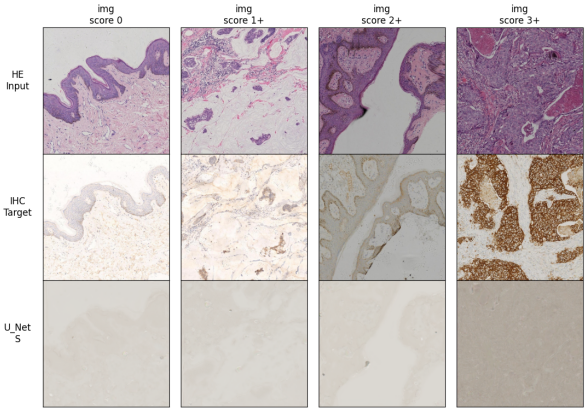


Figure 47: U-Net S generated images

The classifier has an accuracy on the scoring of 0.448 on the images generated by the small U-Net

IHC-score	precision	recall	f1-score	support+
score: 0	0	0	0	30
score: 1+	0.75	0.026	0.05	115
score: 2+	0.445	1	0.616	221
score: 3+	0	0	0	134

Table 39: classification report on U-Net S generated images

7.2.2 U-Net M

The following plot is a sample of the the images generated by the medium U-Net

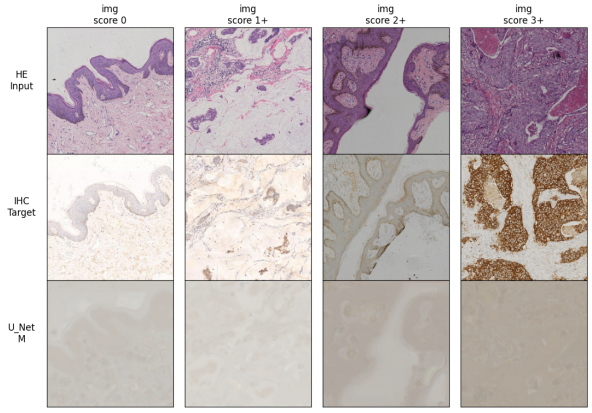


Figure 48: U-Net M generated images

The classifier has an accuracy on the scoring of 0.422 on the images generated by the medium U-Net.

IHC-score	precision	recall	f1-score	support+
score: 0	0.25	0.033	0.058	30
score: 1+	0.296	0.069	0.112	115
score: 2+	0.432	0.914	0.587	221
score: 3+	0	0	0	134

Table 40: Classification report on U-Net M generated images

7.2.3 U-Net L

The following plot is a sample of the the images generated by the large U-Net.

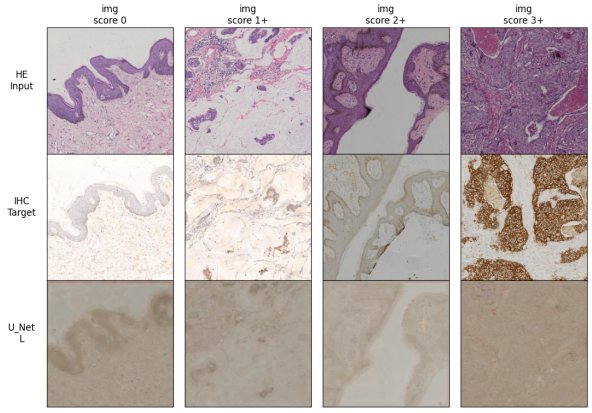


Figure 49: U-Net L generated images

The classifier has an accuracy on the scoring of 0.44 on the images generated by the large U-Net.

IHC-score	precision	recall	f1-score	support+
score: 0	0	0	0	30
score: 1+	0	0	0	115
score: 2+	0.440	0.995	0.611	221
score: 3+	0	0	0	134

Table 41: Classification report on U-Net L generated images

7.2.4 ViT S

The following plot is a sample of the the images generated by the small Vision Transformer.

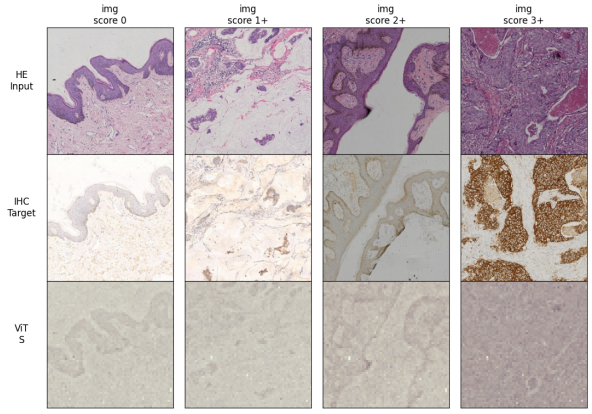


Figure 50: ViT S generated images

The classifier has an accuracy on the scoring of 0.448 on the images generated by the small Vision Transformer.

IHC-score	precision	recall	f1-score	support+
score: 0	0.33	0.066	0.11	30
score: 1+	0.625	0.0434	0.081	115
score: 2+	0.446	0.981	0.613	221
score: 3+	0	0	0	134

Table 42: Classification report on ViT S generated images

7.2.5 ViT M

The following plot is a sample of the the images generated by the medium Vision Transformer.

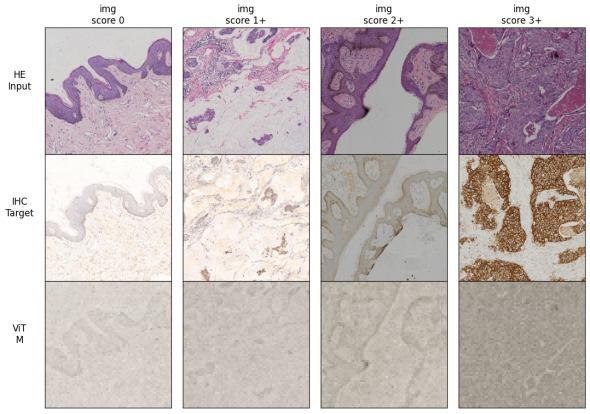


Figure 51: ViT M generated images

The classifier has an accuracy on the scoring of 0.432 on the images generated by the mid sized Vision Transformer.

IHC-score	precision	recall	f1-score	support+
score: 0	1	0.033	0.06	30
score: 1+	0.09	0.008	0.015	115
score: 2+	0.446	0.959	0.609	221
score: 3+	0.15	0.014	0.0272	134

Table 43: Classification report on ViT M generated images

7.2.6 Swin Transformer

The following plot is a sample of the the images generated by the Swin Transformer.

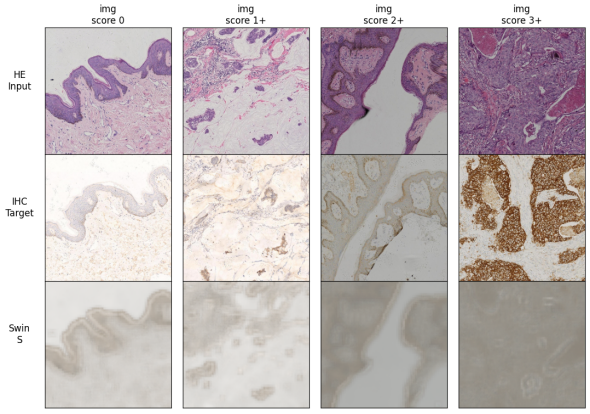


Figure 52: Swin transformer generated images

The classifier has an accuracy on the scoring of 0.46 on the images generated by the Swin Transformer.

IHC-score	precision	recall	f1-score	support+
score: 0	1	0.1	0.18	30
score: 1+	0.38	0.252	0.30	115
score: 2+	0.468	0.886	0.61	221
score: 3+	0.5	0.014	0.02	134

Table 44: Classification report on Swin transformer generated images

7.2.7 Pix2Pix U-Net

The following plot is a sample of the the images generated by the Pix2Pix U-Net.

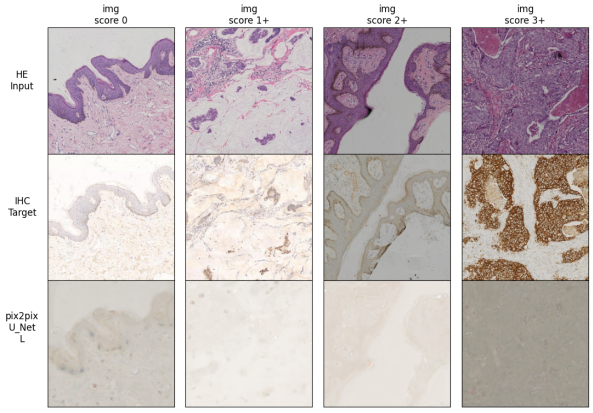


Figure 53: Pix2Pix U-Net generated images

The classifier has an accuracy on the scoring of 0.442 on the images generated by the Pix2Pix U-Net.

IHC-score	precision	recall	f1-score	support+
score: 0	0	0	0	30
score: 1+	0.5	0.008	0.017	115
score: 2+	0.441	0.995	0.611	221
score: 3+	0	0	0	134

Table 45: Classification report on Pix2Pix U-Net generated images

7.2.8 Pix2Pix ViT

the following plot is a sample of the the images generated by the Pix2Pix ViT.

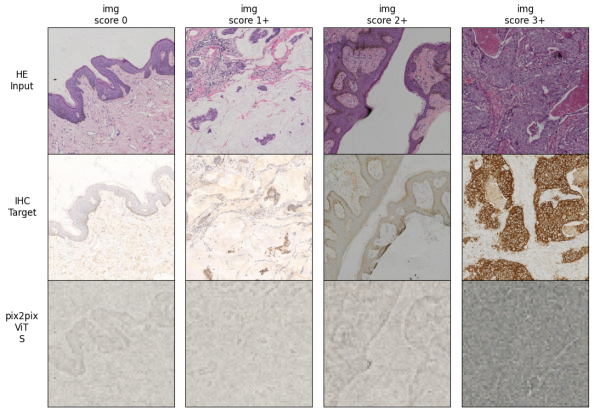


Figure 54: Pix2Pix ViT generated images

The classifier has an accuracy on the scoring of 0.438 on the images generated by the Pix2Pix ViT.

IHC-score	precision	recall	f1-score	support+
score: 0	0	0	0	30
score: 1+	0.5	0.008	0.017	115
score: 2+	0.443	0.986	0.608	221
score: 3+	0	0	0	134

Table 46: Classification report on Pix2Pix ViT generated images

7.2.9 Pix2Pix Swin

The following plot is a sample of the the images generated by the Pix2Pix Swin.

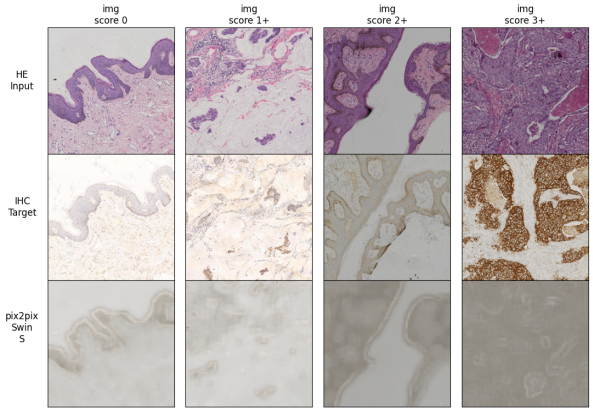


Figure 55: Pix2Pix Swin generated images

The classifier has an accuracy on the scoring of 0.446 on the images generated by the Pix2Pix Swin.

IHC-score	precision	recall	f1-score	support+
score: 0	1	0.066	0.125	30
score: 1+	0.31	0.21	0.25	115
score: 2+	0.467	0.873	0.608	221
score: 3+	0.50	0.022	0.042	134

Table 47: Classification report on Pix2Pix Swin generated images

7.2.10 Diffusion Model

The following plot is a sample of the the images generated by the Diffusion Model.

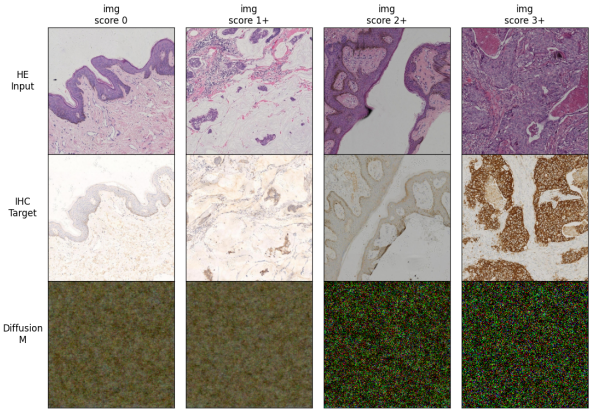


Figure 56: Diffusion Model generated images

Due to the long sampling time and the low quality results in the visual inspection, the images of the Diffusion Model were not classified.

8 Interpretation of Results

In this chapter the quantitative and the qualitative results are discussed for each architecture. As well as some possibilities on further research on this topic.

8.1 Interpretation of Quantitative Results

The task of transferring images from HE-stained to HER2 receptors stained IHC-stained images proves to be inherently challenging. The quantitative analysis does not yield a clear preferred architecture for transferring HE-stained images into IHC-stained images. While the Swin Transformer shows the overall lowest MSE and highest PSNR values, the U-Net L scores the best in the SSIM. The performance from all architectures is better on the test set than on the train set, although the distribution of the IHC-scores among both sets is nearly identical:

IHC-score	train	test
0	6%	6%
1+	23%	24 %
2+	44%	43%
3+	27%	27%

Table 48: Proportion of the IHC-score in he whole set

A further analysis of the two sets shows a possible reason for this discrepancy. The following table shows the MSE, SSIM, and PSNR between the HE-stained image and the IHC-stained images in both the test and the train set:

metric	set	total	score:0	score:1+	score:2+	score:3+
MSE	test	0.0562	0.0458	0.0787	0.0561	0.0392
	train	0.0689	0.0494	0.0887	0.0649	0.0618
$SSIM$	test	0.3993	0.4528	0.3974	0.3963	0.394
	train	0.3101	0.3589	0.3142	0.2974	0.3175
$PSNR$	test	14.3722	16.3222	13.7166	13.9422	15.2072
	train	12.9835	14.7483	12.217	12.6335	13.862

Table 49: Metric scores between HE-stained and IHC-stained images

As seen in 49, the IHC-stained images in the test set are closer to their HE-stained counterparts. This discrepancy can also bee seen in the Box-Whisker-Plots where especially in the analysis of the MSE of all nets the train set shows a significantly bigger number of outliers. This makes the translation task harder and could explain the performance difference in the two sets. A different split of the data could be a solution to this problem in this survey the split was used which is provided in the BCI challenge [8].
The IHC-score of the image seems to have a large impact on the difficulty of the

task. As seen in the Box-Whisker-Plots images with higher IHC-scores make the networks less consistent in the prediction. This is as well shown in the increasing variance of the metrics with a higher IHC-score. This suggests that generating the deep brown patches, in which the staining identified the HER2 protein, is particularly difficult.
The loss graphs shown in the appendix show still a slowly increase in the performance of the networks on the train set. This suggests that a training period of 100 epochs might be to short for yielding the best possible performance of the selected architectures.

8.2 Interpretation of Qualitative Results

The visual inspection shows that most networks generate the basic structure of the the IHC-stained image with a low IHC-score. But they struggle with generating the deep brown patches where the HER2 protein is identified. This reinforces the findings in the quantitative evaluation that all network architectures struggle with the generation of IHC-stained images with a high IHC-score. Especially images with IHC-score 3+ in which changes the structure of the IHC-stained images differs from the HE-stained images perform badly.
The generated images all exhibit a considerably lower contrast than the original IHC-stained images. The low contrast could be a reason why the evaluation of the generated images with the IHC-score classifier did not yield satisfying results. The classifier predicts IHC-score 2+ for almost all generated images because the HER2 expression is not generated detailed enough.

8.3 Interpretation of Architecture Based Results

- **U-Net based architectures:** The analysis of U-Net variations highlights the necessity of the L version, specifically employing a 5-step structure with 64 features. Lighter U-Nets lack the complexity required for generating intricate IHC images. Qualitative assessment reveals that U-Net generated images exhibit low contrast. However, the high SSIM scores suggest that the structural elements are present. Addressing this discrepancy may require the development of an effective post-processing technique to enhance overall image quality.

- **Transformer based architectures:** Visual Transformers with a [4,4] patch size face difficulties in generating high-resolution images. To avoid the computational costs that even smaller patches bring the Swin Transformer was used to address the resolution but the resulting images are blurry. These problems of the Transformer based architectures show in the visual inspection and in the lower SSIM scores. But the low MSE and high PSNR scores suggest that further exploration of these models for IHC image generation is appropriate.

- **Pix2Pix Framework:** The integration of a GAN framework into the architectures appears to have a minimal impact. In neither U-Net based or Transformer based architectures benefit from the additional discriminator loss. The quantitative results are show a slight decreased performance of all networks and in the visual inspection no recognizable difference can be detected. As seen in this survey the Pix2Pix framework is not a good addition to this specific image generation task.
- **Diffusion Model:** The Diffusion Model did not produce the desired results. In the quantitative analysis it shows the worst performance and the visual evaluation shows that none out of the four images resembles the ground truth. These findings along with the long sampling and training time shows that Diffusion Models might not be appropriate for this task.

8.4 Conclusion

Among the evaluated architectures, the U-Net L version demonstrates the best performance. With the highest SSIM score and the most detailed images it is showcasing its effectiveness in achieving the desired image transfer. However, the limitations of the generator become apparent when the ground truth image exhibits a high HER2 expression. To transfer this information onto the generated image is still a limitation on all investigated architectures. That is also apparent in the struggle of the classifier to accurately predict IHC-scores on the generated images. Consequently, a more detailed qualitative analysis becomes imperative, with pathologists playing a crucial role in assessing the quality of the generated images and in developing a post processing.

8.5 Future Work

In pursuit of developing a network that can reliably transfer HE-stained images to IHC-stained images the work in this survey can be extended. The low contrast of the generated images could be improved with the implementation of a post processing of the generated images.
To gain more insight from the visual inspection of the generated images a evaluation by medical experts is necessary.
Another approach could be to develop a training in a Pix2Pix framework where the discriminator is predicting the IHC-score of the generated image instead of between real and fake IHC-stained images.

9 Appendix

9.1 Loss-graphs

9.1.1 U-Netbased architectures

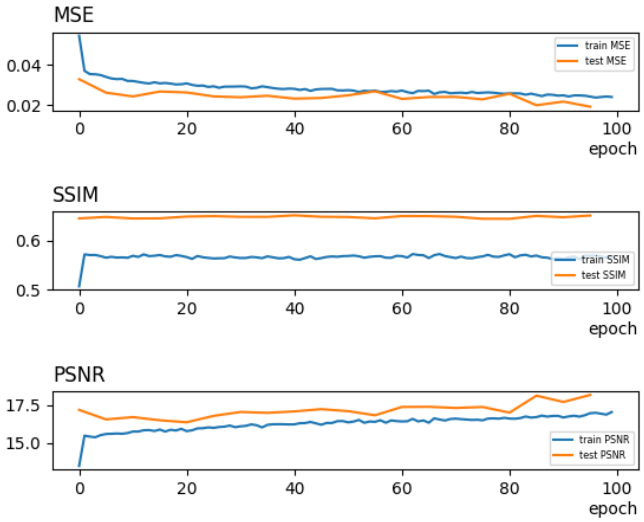


Figure 57: Train and Test loss for "U-Net S"

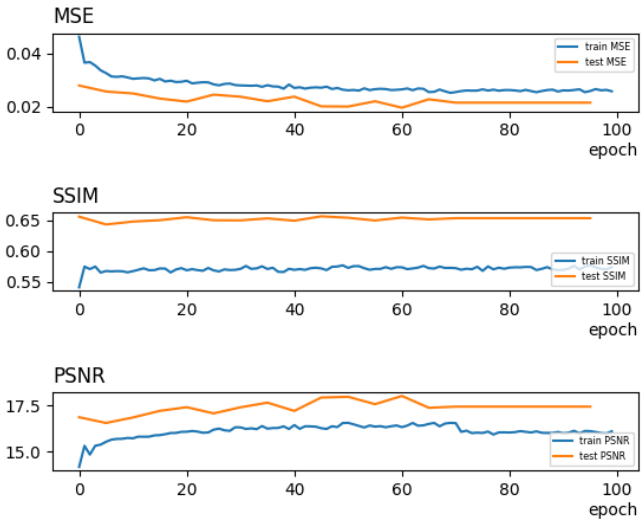


Figure 58: Train and Test loss for "U-Net M"

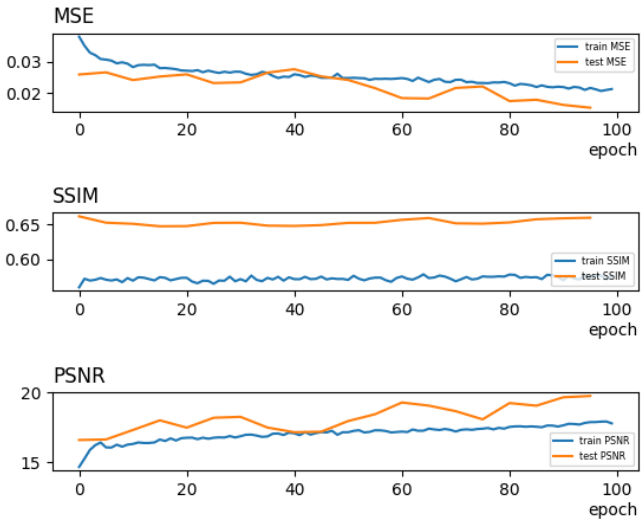


Figure 59: Train and Test loss for "U-Net L"

9.1.2 Transformer based architectures

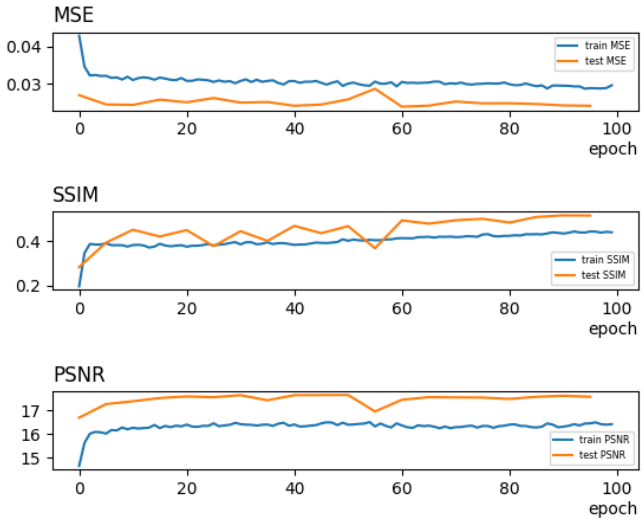


Figure 60: Train and Test loss for "ViT S"

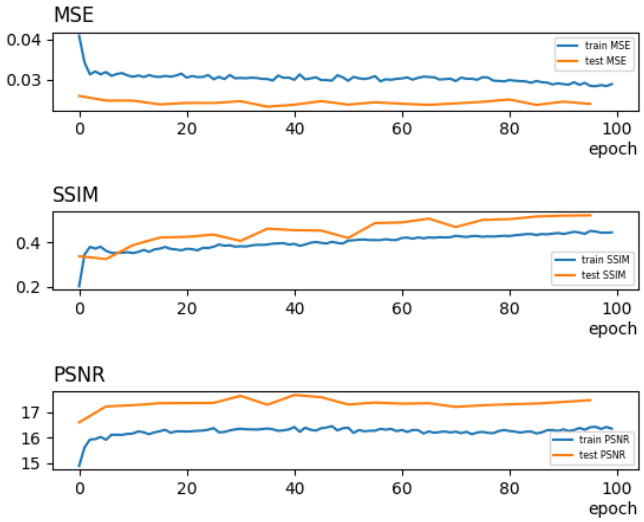


Figure 61: Train and Test loss for "ViT M

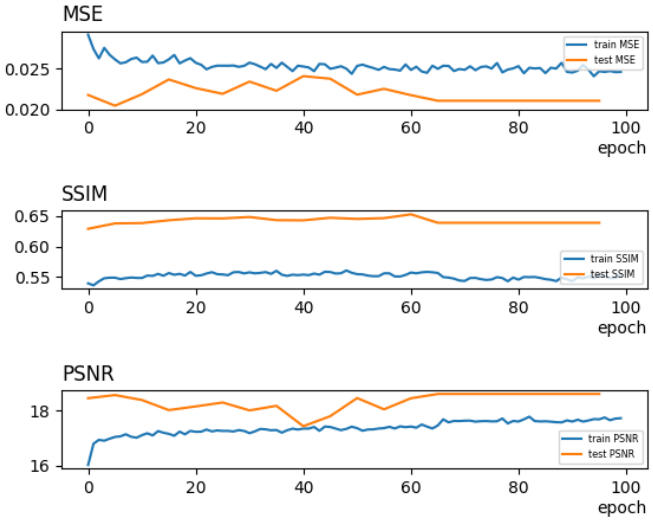


Figure 62: Train and Test loss for "Swin T"

9.1.3 Pix2Pix based architectures

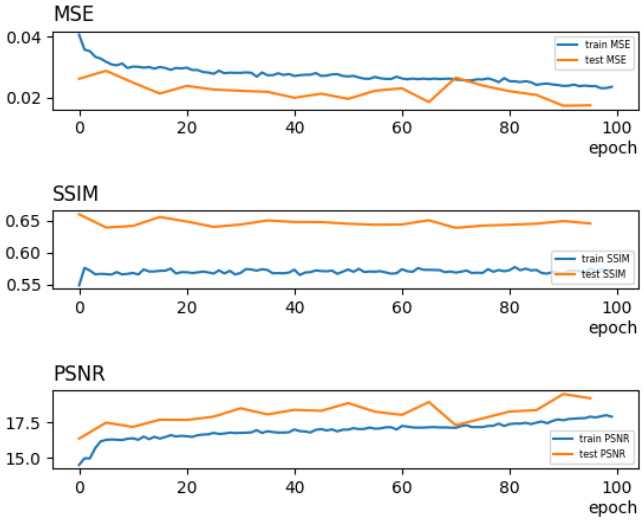


Figure 63: Train and Test loss for "pix2pix U-Net"

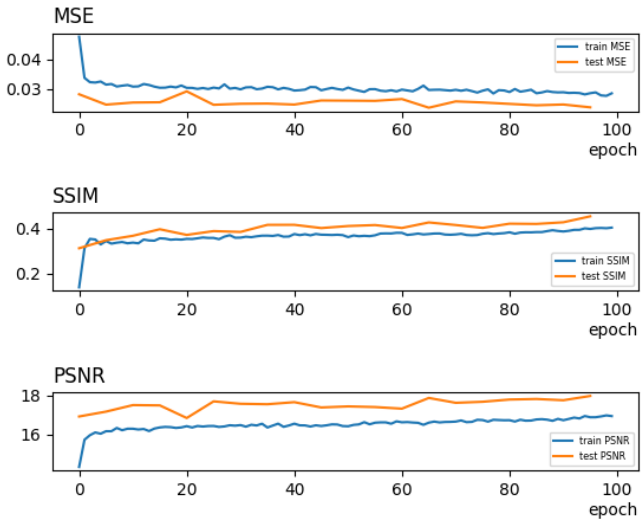


Figure 64: Train and Test loss for "pix2pix ViT S"

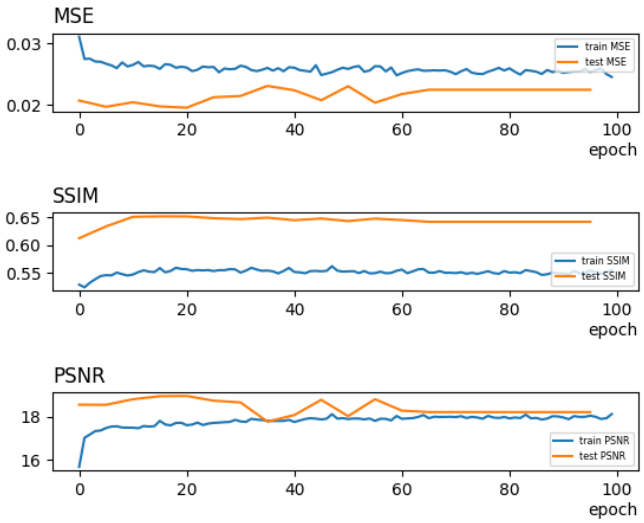


Figure 65: Train and Test loss for "pix2pix Swin Transformer"

9.2 Confusion matrix

9.2.1 U-Net based architectures

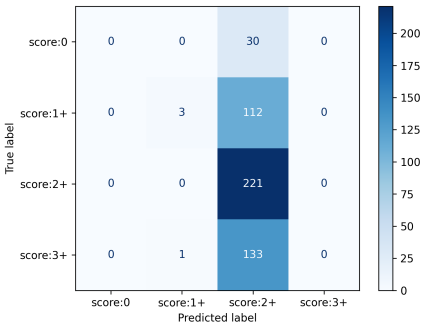


Figure 66: Confusion matrix for the classifier on the generated images by U-Net s

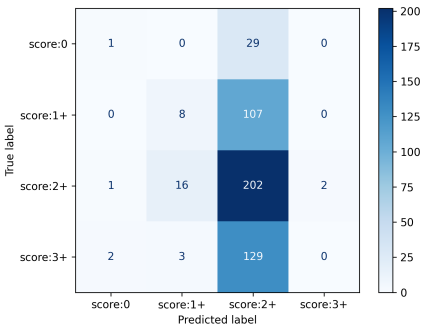


Figure 67: Confusion matrix for the classifier on the generated images by U-Net M

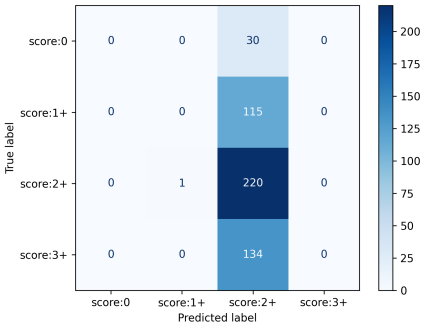


Figure 68: Confusion matrix for the classifier on the generated images by U-Net L

9.2.2 Transformer based architectures

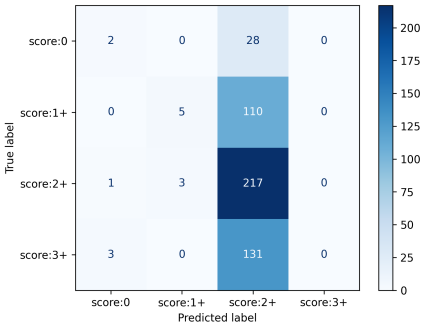


Figure 69: Confusion matrix for the classifier on the generated images by ViT S

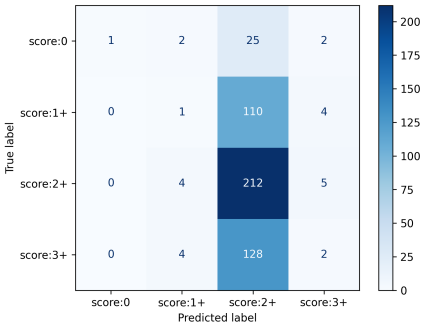


Figure 70: Confusion matrix for the classifier on the generated images by ViT M

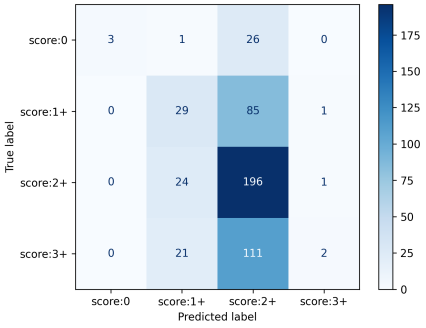


Figure 71: Confusion matrix for the classifier on the generated images by Swin transformer

9.2.3 Pix2Pix based architectures

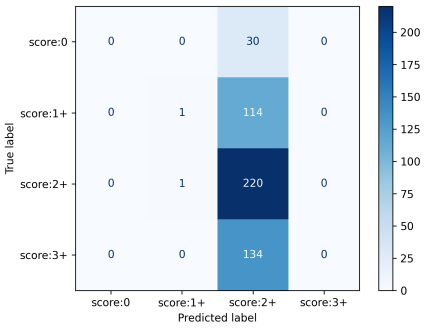


Figure 72: Confusion matrix for the classifier on the generated images by Pix2Pix U-Net

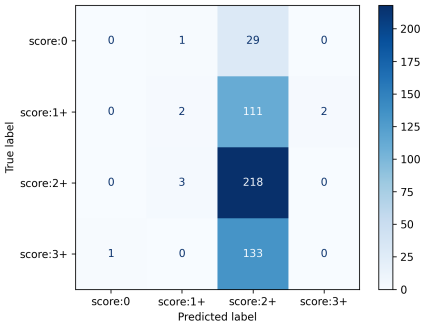


Figure 73: Confusion matrix for the classifier on the generated images by Pix2Pix ViT

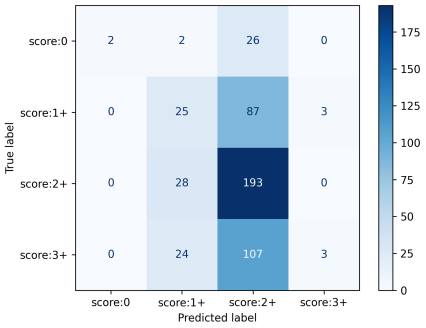


Figure 74: Confusion matrix for the classifier on the generated images by Pix2Pix Swin

List of Figures

List of Tables

References

[1] Albert H Coons, Hugh J Creech, and R Norman Jones. Immunological properties of an antibody containing a fluorescent group. *Proceedings of the society for experimental biology and medicine*, 47(2):200–202, 1941.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[6] Saidul Kabir, Semir Vranic, Rafif Mahmood Al Saady, Muhammad Salman Khan, Rusab Sarmun, Abdulrahman Alqahtani, Tariq O Abbas, and Muhammad EH Chowdhury. The utility of a deep learning-based approach in her-2/neu assessment in breast cancer. *Expert Systems with Applications*, 238:122051, 2024.

[7] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[8] Shengjie Liu, Chuang Zhu, Feng Xu, Xinyu Jia, Zhongyue Shi, and Mulan Jin. Bci: Breast cancer immunohistochemical image generation through pyramid pix2pix. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1815–1824, June 2022.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image*

Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

[11] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022.