ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Neural Style Transfer for Ultrasound Imaging

Semester's Thesis

Davide Menini
Department of Information Technology and Electrical Engineering (D-ITET)

Advisors: Lin Zhang, Dr. Tiziano Portenier
Supervisor: Prof. Orcun Goksel

July 20, 2020

Abstract

The simulation of ultrasound images is fundamental for the training of specialized medical personnel. This semester thesis aims to develop a post-processing step to improve quality and realism of simulated US images by using Neural Style Transfer. The idea is to transfer an high-quality US style or a clinical US style, depending on the task, on a simulated content image. We employ an optimization approach, which iteratively minimizes a weighted combination of style and content loss, and a learning approach, which consists of a neural network trained with a perceptual loss. Moreover, we enrich these algorithms with some additions to take advantage of all the information available as input, e.g. segmentation maps and a whole dataset of style images. Eventually, the outcomes show that the learning approach outperforms the optimization approach. Indeed the output images better represent the input style, and the network runs in real-time on GPU. Unfortunately, the best result only achieves a SSIM of 70% on average, which is not optimal but is a good starting point for further research.

Acknowledgements

I would like to express my deep gratitude to Professor Orcun Goksel, who offered me the opportunity to work in the amazing laboratory that CVL is. Unfortunately this period has been peculiar for everyone, but the extreme expertise of CVL members and the infrastructures provided by ETH helped to relieve the difficulty of working from home. Hence, I feel obliged to extend my thanks to everyone who has made my life easier.
I would also like to offer my special thanks to my supervisors Lin Zhang and Tiziano Portenier for their patient guidance and useful critiques of this research work.
Finally, I wish to thank my parents and friends for their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Medical ultrasound (US) is one of the most widely used medical imaging techniques for its radiation-free, affordable, portable, and real-time nature. However, an extensive training of radiologists is required to face challenges as probe navigation and image interpretation. Traditional training with synthetic phantoms is often unrealistic; and training with cadavers or patients involves ethical issues. In contrast, model-based numerical US simulation can allow for training in virtual reality, enabling to artificially recreate complex medical scenarios and rare pathologies without any risk. Thus, developing US sonograms simulators which can produce images indistinguishable from real ones in real-time has been a major research interest [19].

US speckle texture is the interference pattern resulting from echos scattered by sub-wavelength tissue structures (*scatterers*). It is of major importance, as it provides tissue-specific descriptive and diagnostic information. US texture can be faithfully modeled by the convolution of a tissue representation in the form of parametrized point scatterers, with a spatially varying point spread function. Anyway, in order to achieve a realistic texture appearance, a viable abstract tissue representation (*scatter map*) is necessitated.

State-of-the-art methods for extracting scatter maps use optimization approaches. Another solution proposed by Al Bahou *et al.* [1] uses a learning approach based on the *pix2pix* [7] Conditional Generative Adversarial Networks (CGANs), which are GANs [5] that allow synthesizing new images conditioned by the characteristics of the input (image-to-image translation). Eventually, the simulated images are quite accurate but still not completely realistic. Successful attempts to improve realism on abdominal ultrasound simulations [13] have been achieved by using GANs trained with a cycle consistency loss (CycleGANs), which allow a dataset of unpaired images to be used for training [18].

1.1 Focus of this Work

In this semester thesis we propose a post-processing step to further improve the quality and realism of simulated images by means of Neural Style Transfer [4]. The idea is to consider the real and

simulated images as two different styles and transfer the "real" or the "high-quality" style to the content of the simulated images. We consider two main approaches:

- The original Neural Style Transfer optimization problem proposed by Gatys *et al.* [4], and its augmentation with the content semantic information, as developed by Johnson *et al.* [10].
- A real-time Neural Style Transfer based on a feed-forward network trained with a perceptual loss, designed by Luan *et al.* [8].

Our analysis is performed on a dataset of simulated ultrasound images, generated via GANs from different fetal models [12]. In addition, we also use a dataset of clinical ultrasound images of fetal heads [16] provided by the "Automated Measurement of Fetal Head Circumference" challenge [15].

To the best of our knowledge, this is the first attempt of performing Neural Style Transfer on simulated ultrasound images to improve the realism of the outcome.

1.2 Thesis Organization

In order to understand the theoretical background, in Chapter 2 we present an overview of the papers that mostly influenced our work. In Chapter 3 we discuss our datasets and ideas, which lead to the results shown in Chapter 4 and eventually discussed in Chapter 5.

Chapter 2

Theoretical Background

2.1 Neural Style Transfer

Neural Style Transfer [4] is an optimization technique used to apply the style of a *style reference image* on the content of a *content image*. This is implemented by optimizing the output image (*stylized image*) to match both the content features of the content image and the style features of the style image at the same time. These representations are extracted from the images using a Convolutional Neural Network (CNN), which for our purpose is the VGG19 network architecture [14], a pretrained classification model trained with the ImageNet dataset [3].

The intermediate layers of VGG19 are used to get the content and style representations of the image. The first few layers of the network represent low-level features like edges and textures, while the final layers represent higher-level features like eyes or wheels. To extract the style information we are interested in both levels of features, while to extract the content information only the high-level features are useful. Of course, the choice of the feature extractor network is important: a better model allows to extract better features, which in turn means better stylized images.

Once the target content representation C and target style representation S are extracted, they need to be matched by the output image O via an optimization algorithm (gradient descent), which minimizes the weighted sum of content and style losses:

$$\mathcal{L} = \sum_{\ell=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_s^\ell \tag{2.1}$$

where L is the total number of extracted layers; α and β are the content and style weights respectively.

The content loss is the mean squared error of the target content features $F_\ell[C]$ and the output content features $F_\ell[O]$, where $F_\ell[\cdot] \in \mathbb{R}^{M_\ell \times N_\ell \times C_\ell}$ is the feature map extracted via VGG19 from the content or output images:

$$\mathcal{L}_c^\ell = \frac{1}{2M_\ell N_\ell C_\ell} \sum_{i,j,k} (F_\ell[C] - F_\ell[O])_{ijk}^2 \tag{2.2}$$

The style of an image can be described by the means and correlations across the different feature maps. This information is comprised in the Gram matrix of the features. The Gram matrix for a particular layer G_ℓ can be calculated by taking the outer product of the feature vector with itself at each location, and averaging over all locations:

$$G_\ell(c,d) = \frac{1}{M_\ell N_\ell} \sum_{i,j} F_\ell(i,j,c) \cdot F_\ell(i,j,d) \tag{2.3}$$

The style loss is then defined as the mean squared error of the Gram matrix of the target style features, and the Gram matrix of the output style features:

$$\mathcal{L}_s^\ell = \frac{1}{2C_\ell^2} \sum_{i,j} (G_\ell[S] - G_\ell[O])_{ij}^2 \tag{2.4}$$

2.2 Augmented Style Loss

In [10], Johnson *et al.* develop a new version of the style loss to compensate the fact that the Gram matrix is computed over the entire image. Indeed, this limits its ability to adapt to variations of the semantic context and can cause "spillovers". They address this problem by generating image segmentation masks for common labels of the style and content images, which are added to the content image as additional channels. Moreover, the neural style algorithm is augmented by updating the style loss as follows:

$$\mathcal{L}_{s+}^\ell = \sum_{d=1}^{D} \frac{1}{2C_{\ell,d}^2} \sum_{i,j} (G_{\ell,d}[S] - G_{\ell,d}[O])_{ij}^2 \tag{2.5}$$

$$F_{\ell,d}[S] = F_\ell[S] M_{\ell,d}[S] \qquad F_{\ell,d}[O] = F_\ell[O] M_{\ell,d}[C]$$

where D is the number of channels in the semantic segmentation mask (considering a channel for each label), $M_{\ell,d}$ denotes the channel d of the segmentation mask in layer ℓ, and $G_{\ell,d}[\cdot]$ is the Gram matrix corresponding to $F_{\ell,d}[\cdot]$. The masks are downsampled to match the feature map spatial size at each layer.

2.3 Perceptual Loss Network

Perceptual losses allow to capture perceptual differences between output and ground-truth images, i.e. differences between high-level feature representation extracted from a pretrained CNN. Johnson *et al.* [8] train feedforward transformation networks for image transformation tasks by using perceptual loss functions that depend on high level features extracted from a pretrained VGG16. In particular, they train an *Image Transform network* f_W, which is an encoder-decoder convolutional neural network that transforms the input image x (content image) into the output image $\hat{y}$ (stylized image) through the mapping $\hat{y} = f_W(x)$. The output image $\hat{y}$, the content target image $y_c = x$

and the style target image y_s are then fed to the *loss network* ϕ (pretrained VGG16), which is used to extract high-level features from the inputs and calculate the content loss (Eq. 2.2) and the style loss (Eq. 2.4). This architecture is depicted in Figure 2.1. At test-time the transformation network runs in real-time, allowing for real-time style transfer.

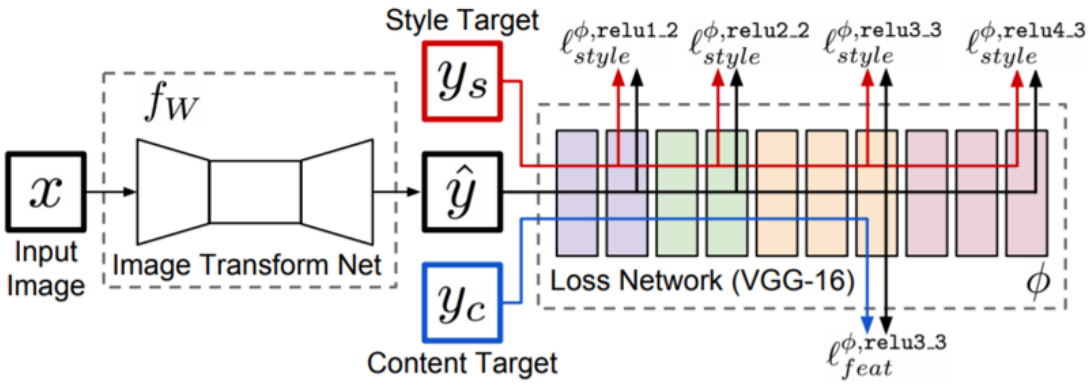

Figure 2.1: Perceptual loss network architecture [8]

Chapter 3

Materials and Methods

3.1 Datasets

The main dataset is composed of 6669 abdomen ultrasound simulated images of feti. These US images are generated using a Monte-Carlo ray tracing framework to render B-mode images from a custom geometric fetal model for obstetric training [12]. As shown in Figure 3.1, each image in the dataset contains three sub-images of size 1000x1386 representing the same content in a different "style", which from left to right is (1) simulated Low-Quality (LQ) US image, (2) simulated High-Quality (HQ) US image, (3) segmentation map. Paired LQ and HQ images are generated using two simulation passes with different number of primary rays per US scanline and number of elevational layers [12]. This dataset is used for two tasks:

- *Low-Quality to High-Quality* style transfer
- *Segmentation to High-Quality* style transfer

The second dataset is composed of 800 clinical ultrasound images of fetal heads [16] provided by the "Automated Measurement of Fetal Head Circumference" challenge [15]. Here the images are screenshots directly taken from ultrasound scanners. On the left there is a brightness bar which has to be removed by a preprocessing step. See Figure 3.2 for an example image.
This clinical dataset, along with the simulated dataset, is used for the following task:

- *High-Quality to Clinical* style transfer

3.2 Neural Style Transfer

The first approach is the basic Neural Style Transfer optimization problem (cf. Section 2.1) by Gatys et al. [4]. The Adam optimizer is used to minimize a weighted combination of loss functions:

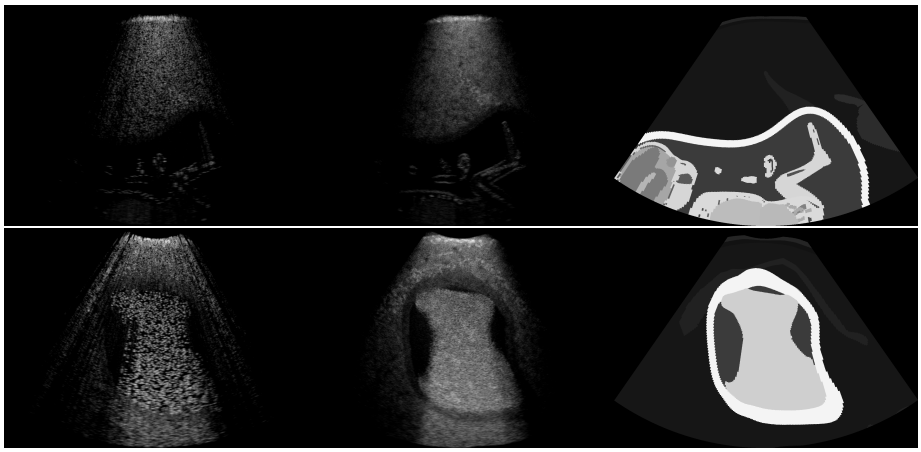

Figure 3.1: Example images from CVL simulated dataset (`34.png`, `18.png`): LQ image (left), HQ image (center), segmentation map (right).

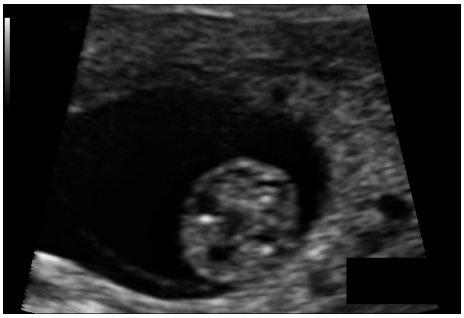

Figure 3.2: Example image from clinical dataset (`000_HC.png`).

- The *content loss* is the Euclidean distance between feature representations (Eq. 2.2). Since the content features are chosen among high-level layers of VGG19, this loss is minimized for high-level similarities between the output image and the content image. This means that the content and the spatial structure are preserved, while the colour, exact shape, texture are not.
- The *style loss* is the squared Frobenius norm of the difference between the Gram matrices of the feature representations (Eq. 2.4). This loss is minimized for similarities in style (colors, textures, common patterns) between the output image and the style image. Spatial structure and content information are not preserved.
- The *total variation loss* is a regularization term that penalizes the high frequency components of the image. In some application like feature inversion [11] or image super-resolution [2] it is helpful, but its smoothing effect is unwanted in our application.

A modification to this first approach uses an Augmented Style Loss based on the one proposed by Johnson *et al.* [10], presented in Section 2.2. From the segmentation map of the content, a binary mask for each label is extracted. This means that depending on the content, the masks range from 10 to 40, leading to some problems with the GPU memory and the optimization duration. This overhead is reduced by considering only the labels related to the main object (i.e. the fetus), whose values are in the interval $[150, 220]$. As additional filtering, we consider only labels that generate a "significant" mask, i.e. a mask whose white area is greater than the 0.25% of the image. Figure 3.3 shows some masks obtained from the segmentation map from image `34.png`. The following tables show the amount of labels and the actually used labels after filtering for a subset of images, which are generated using the same fetus surface model, but different imaging planes:

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Tot Labels	19	15	13	12	12	10	9	9	9	9	9	9	9	9	9	10	10
Used Labels	4	3	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1

Image	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Tot Labels	10	10	9	10	10	10	10	10	10	12	11	12	14	41	35	34	26
Used Labels	1	1	1	1	1	1	1	1	1	2	2	3	3	7	6	6	5

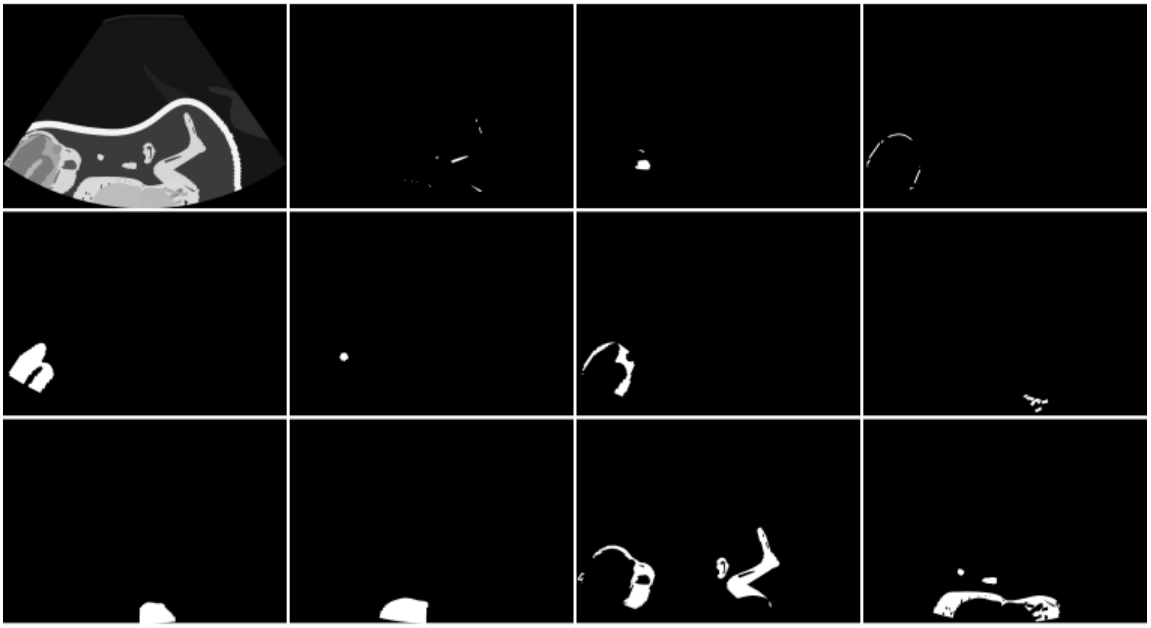

Figure 3.3: Subset of the masks obtained from the segmentation map (top left image).

Additionally, we also consider a mask for the whole object, obtained as the sum of the previous masks, and a mask for the foreground, useful to clear the background noise as post-optimization step.

Once the masks are generated, they are resized to the the size of the style features by subsequent fake operations that resemble the architecture of VGG19. In detail:

- `fake_conv` is a 3×3 average operation that resembles the VGG19 3×3 convolution.
- `fake_pool` is a nearest neighbour halving operation that resembles the 2×2 max pool.

Eventually, these resized masks are used as input for the augmented style loss. For the "Low-quality to High-quality Transfer" and "Segmentation to High-quality Transfer", where the segmentation map for the style image is available, we can apply the same loss presented by Johnson *et al.* (cf. Eq. 2.5). To avoid problems between unpaired labels, we consider only the ones in common between content and style maps. However, in a more general scenario the style segmentation could not be available, thus the augmented style loss assumes the following expression, where D is the total number of used masks:

$$\mathcal{L}_{s+}^{\ell} = \sum_{d=1}^{D} \frac{1}{2C_{\ell,d}^2} \sum_{i,j} (G_{\ell,d}[S] - G_{\ell,d}[O])_{ij}^2 \tag{3.1}$$

$$F_{\ell,d}[S] = F_{\ell}[S]M_{\ell,d}[C] \qquad F_{\ell,d}[O] = F_{\ell}[O]M_{\ell,d}[C]$$

In the previous solution the masking happens in the feature space. Another idea is to perform the masking in the image space and transfer the desired style locally: each mask is used to extract only a certain region of the content image, which is then *locally* optimized. The final image is obtained as the sum of the different local optimizations.

Finally, another variation from the basic algorithm was done while seeking to exploit the whole information available as input. In particular, our datasets are composed of many images representing the same style (i.e. US HQ style or US clinical style) in a different manner. From these style images, we create an *average style* by averaging over (1) the feature space, and (2) the Gram space (both solutions have been tried for the sake of completeness). Eventually, the obtained average style is used as target for the optimization process as usual.

3.3 Perceptual Loss Network

The Image Transform network is trained on the content dataset and it learns to transfer the style of the style image (i.e. HQ or clinical, depending on the task) by minimizing a weighted combination of content and style losses, computed on high-level features extracted via VGG19. The implementation is almost the same as the original one [8], introduced in Section 2.3, the main difference being the usage of VGG19 instead of VGG16 as feature extractor. In addition, the original paper employs also the total variation loss, which we discard for its undesired smoothing effect.

Moreover, we modified the Image Transform Net by introducing some noise layers in the decoding section of the network, as similarly presented in StyleGAN [9]. These layers consist of weighted gaussian noise, whose weighting can be learnt by the network, and are active both at

training and test time. The injection of noise in the decoding layers helps to create local stochastic variations when needed and it affects only the random aspects, leaving the overall content and high-level characteristics unaffected.

Eventually, we also try to train the network by using an average style calculated over the feature representations as target for the style loss, with the aim of making the training independent on the actual chosen style image.

Chapter 4

Results

4.1 Metrics

In general, neural style transfer is an ill-posed problem because of the absence of a unique solution. Anyway, in this application we have an ideal target image which is the High-Quality image correspondent to the content. This allows to define some quantitative scores to enforce our analysis: the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity (SSIM) index [17].
The PSNR, expressed in logarithmic decibel scale, is the ratio between the maximum intensity of the target image I and the Mean Squared Error of the target image I and the output image O:

$$PSNR = 20 \log_{10} \frac{max(I)}{\sqrt{MSE(I,O)}} \tag{4.1}$$

The SSIM [17] is a local index calculated on various windows of an image. It considers image degradation as perceived change in structural information, i.e. the idea that nearby pixels have strong inter-dependencies which carry information about the structure of the objects in the visual scene. The index between two windows x and y of size $N \times N$ is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.2}$$

$$c_i = (k_i L)^2 \qquad L = 2^{bits\ per\ pixel} - 1$$

Anyway, these metrics are computable only for the 'LQ to HQ' and 'Seg to HQ' tasks, where a target image is available. Unfortunately, this is not possible for the 'HQ to Clinical' transfer, hence another metric needs to be used to quantify the analysis. For this purpose, we employ the FID, firstly introduced by Heusel et al. [6].
Often used to evaluate GANs, the FID (Fréchet Inception Distance) metric is helpful to compare statistics from different sets of images, as the US clinical images (target) and the stylized US

clinical images (post-optimization). The Fréchet Distance is the distance d between the Gaussian $\mathcal{N}(\mu_1, C_1)$ (synthetic images) and the Gaussian $\mathcal{N}(\mu_2, C_2)$ (real-world images):

$$d^2 = ||\mu_1 - \mu_2||_2^2 + Tr(C_1 + C_2 - 2(C_1 C_2)^{1/2}) \tag{4.3}$$

Anyway, FID needs a lot of samples in order to be accurate. This is only a small issue for the perceptual loss network due to its real-time behaviour, but unfortunately it is not feasible for the basic Neural Style Transfer, because it takes much longer to optimize the same amount of images.

4.2 Neural Style Transfer

The Neural Style Transfer algorithm is implemented in Tensorflow. VGG19 is used as feature extractor: the style features are extracted from layers *block1_conv1*, *block2_conv1*, *block3_conv1*, *block4_conv1*, *block5_conv1*; the content features are extracted from *block5_conv2*. Indeed, as previously mentioned, for the style we use both high-level and low-level features (i.e. colour, edges, texture, etc), while for the content we are only interested in high-level features (i.e. pose, spatial orientation, etc).
Adam optimizer is used to minimize the weighted combination of style and content losses. The learning rate and loss weights are obtained from a grid search whose goal is to maximize the metrics and provide a qualitative good output image. Figure 4.1 shows the optimization outcomes with different learning rates. A value of $5\mathrm{e}{-3}$ gives an output image close to the middle results and also optimal SSIM and PSNR.
Figure 4.2 shows the optimization outcomes for different values of the content weight, keeping the style weight fixed at $1\mathrm{e}{-2}$. As expected, when there is only the style loss (Fig. 4.2.a) the result is a fragmented patchwork of texture regions surrounded by dark areas, without any spatial information at all. Only when the content loss is strong enough (Fig. 4.2.c), the output starts to resemble the content image. Depending on the selected style image, this behaviour may change.

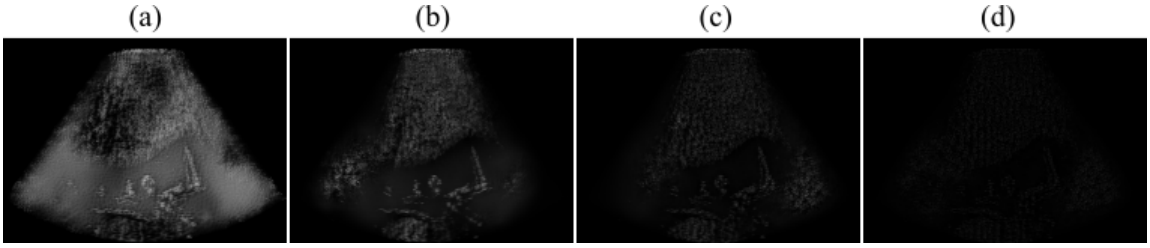

Figure 4.1: Optimization outcomes with different learning rates: (a) 0.1, (b) 0.01, (c) 0.001, (d) 0.0001.

The choice of the loss weights depends on the task, e.g. when the content is a segmentation map the style needs to be stronger in order to balance the losses. Table 4.1 summarizes the weights used for each task and the value of the losses after 3000 iterations.

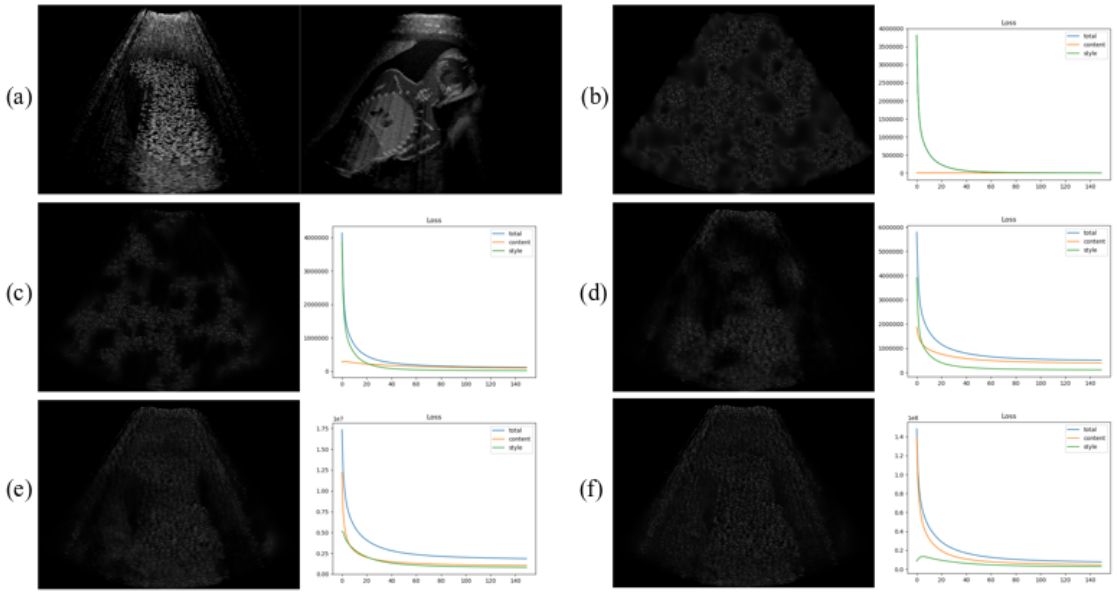

Figure 4.2: Optimization outcomes with different content weights cw, and respective loss plots: (a) content and style images, (b) $cw = 0$, (c) $cw = 1e3$, (d) $cw = 1e4$, (e) $cw = 1e5$, (f) $cw = 1e6$.

Task	Content weight	Style Weight	Learning Rate	Content Loss	Style Loss
LQ to HQ	1e5	1e−2	5e−3	4e5	1e5
Seg to HQ	1e5	1e−1	5e−3	8e6	3e6
HQ to Clinical	5e5	1e−2	5e−3	2e6	2e6

Table 4.1: Loss weights for the different tasks, with the respective losses after 3000 iterations.

Eventually, before showing off the results of the various tasks, it is interesting to discuss the algorithms' profiling. The basic NST takes roughly $600s$ to compute 1500 iterations on Titan X, which are the least amount necessary in order to obtain a satisfying decrease in the loss. Actually, the loss keeps decreasing until 3000 iterations ($1200s$), thus this is the value considered throughout our analysis.
The NST with augmented style loss greatly increases the optimization duration because of the masks. Values show that the time increases linearly with the number of masks: each mask produces roughy $330s$ of overhead over 3000 epochs. Unfortunately, the GPUs may crash with more than 10 masks because of memory allocation problems. Table 4.2 gives more details on the relation between masks and timing.
The local NST is by far the algorithm that takes more time to compute. Indeed, a basic NST optimization is performed for each label of the segmentation map. This means that if we consider 8 masks, 8 consecutive optimizations need to be done, which take $9600s$ in total.

	StyleLoss	StyleLoss+					
# Masks	0	1	2	4	6	8	10
Time (s)	1200	1580	1910	2570	3230	4000	4710
s / iter	0.400	0.527	0.637	0.857	1.077	1.333	1.570

Table 4.2: NST profiling.

Low-quality to High-quality NST
In this task the content is a low-quality image, while the style is a high-quality image. For reference, the content is `34.png`, the style is either the same image for the paired experiment, or `645.png` for the unpaired one. Figure 4.3 shows how NST generalizes the style between optimizations where the content and style are *paired* (a) and *unpaired* (b). Figure 4.4 shows the output images for the different approaches introduced in the previous chapters (cf. Ch. 3.2): (a) Neural Style Transfer with basic style loss, (b) NST with *original* augmented style loss, (c) masking in the image space followed by local NST mask-by-mask, (d) NST with average style over the feature space. In each case, the optimization is performed with unpaired content-style images. Tables 4.3 and 4.4 gives the scores for these comparisons.

Segmentation to High-quality NST
In this task the content is a segmentation map, while the style is a high-quality image. For reference, the content is `34.png`, the style is `645.png`. Figure 4.5 shows the outcomes for an unpaired experiment. Only basic NST and NST with the *original* augmented style loss (Eq. 3.1) are used here. The respective scores are shown in Table 4.5.

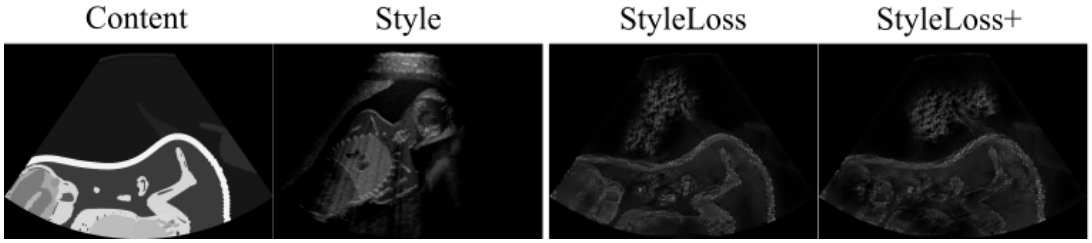

Figure 4.5: Segmentation to HQ images with unpaired content-style.

Seg to HQ	StyleLoss		StyleLoss+	
	SSIM	PSNR	SSIM	PSNR
Unpaired	46.24%	19.21	45.17%	18.86

Table 4.5: Figure 4.5 scores.

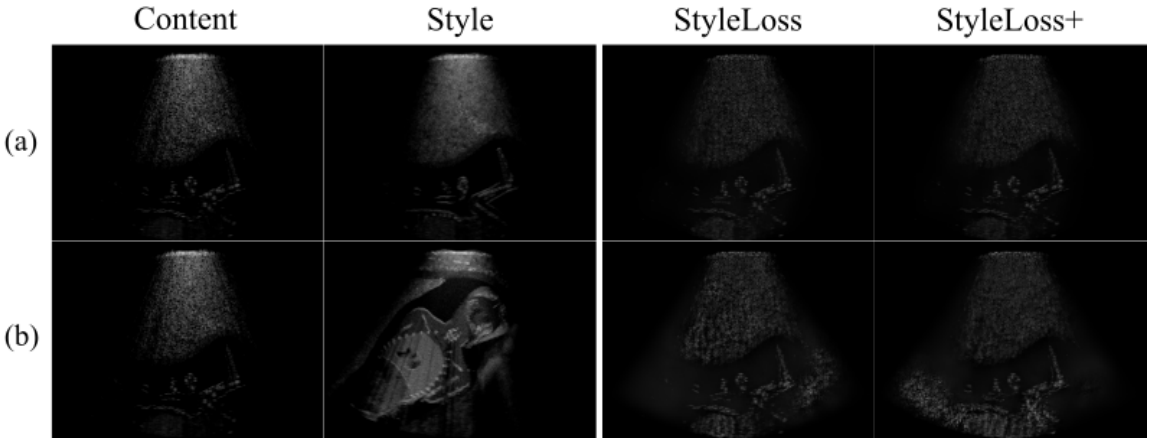

Figure 4.3: LQ to HQ images with (a) paired and (b) unpaired content-style.

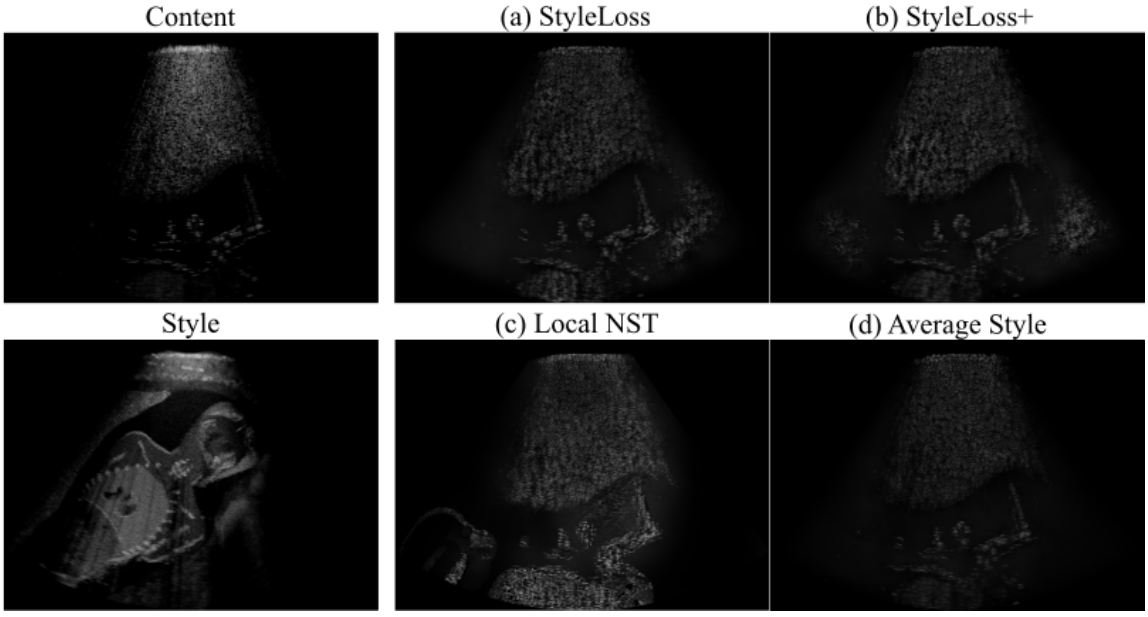

Figure 4.4: LQ to HQ images obtained with different approaches: (a) basic Neural Style Transfer, (b) NST with augmented style loss, (c) Local NST, (d) NST with average style in feature space.

LQ to HQ	StyleLoss		StyleLoss+	
	SSIM	PSNR	SSIM	PSNR
Paired	63.5%	23.72	63.9%	23.77
Unpaired	50.5%	22.56	50.9%	21.86

Table 4.3: Figure 4.3 scores.

LQ to HQ	SSIM	PSNR
StyleLoss	50.5%	22.56
StyleLoss+	50.9%	21.86
Local NST	48.8%	21.19
Average Style	54.8%	23.39

Table 4.4: Figure 4.4 scores.

High-quality to Clinical NST

In this task the content is a high-quality image, while the style is a US clinical image. For reference, the content is `34.png`, the style is `000_HC.png`. Figure 4.6 shows the optimization outcomes for different experiments, namely (a) basic NST, (b) NST with *modified* augmented style loss, (c) NST with average style over the feature space, (d) NST with average style over the Gram space. Since this task does not have any target image, it is not possible to determine the metrics.

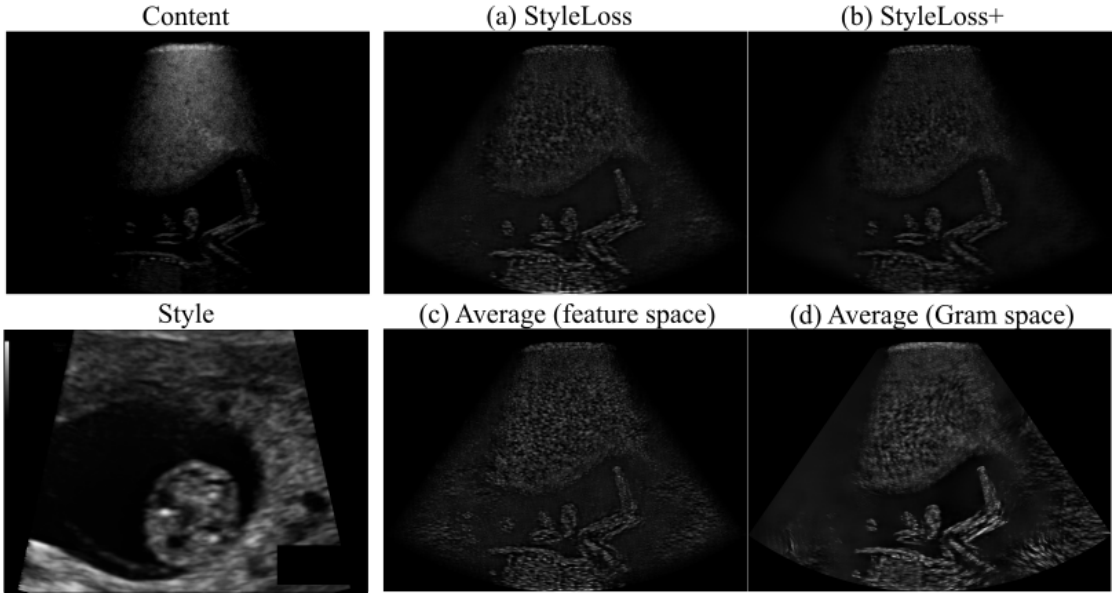

Figure 4.6: HQ to Clinical images.

4.3 Perceptual Loss Network

The Perceptual Loss paper is implemented in Pytorch. The VGG19 layers used to extract the features are *block1_conv1*, *block2_conv1*, *block3_conv1*, *block4_conv1*, *block5_conv1* for the style, and *block5_conv2* for the content. Regarding the training, the learning rate is $1e{-}5$, the batch size is 1 (limited by GPU memory), while the weights vary depending on the task (cf. Table 4.6).

Task	Content weight	Style Weight	Learning Rate	Batch Size	Input size
LQ to HQ	1e2	5e6	$1e{-}5$	1	1000×1000
Seg to HQ	1e2	5e7	$1e{-}5$	1	1000×1000
HQ to Clinical	1e2	5e6	$1e{-}5$	1	1000×1000

Table 4.6: Perceptual loss network training parameters.

Before being fed to the Image Transform Net, the images are center-cropped to size 1000×1000. This does not represent an information loss, because due to the structure of our images, mainly black sections from the sides are removed. Roughly 10% of the images from the provided dataset are used for test, the remaining (6000 images) are used for training. With 1000×1000 of input size, the basic training requires roughly 8 hours to compute 5 epochs. However, with the addition of three noise injection layers in the decoding section of the network, the training time increases to 11 hours. Actually, it could be reduced by decreasing the input shape, although this solution leads to worst results. Despite the "long" training time, style transfer happens almost in real-time, with the outcomes provided below in Figure 4.7 (Low-quality to High-quality NST), Figure 4.8 (Segmentation to High-quality NST) and Figure 4.9 (High-quality to Clinical NST). These figures compare the outputs of the Image Transform Net in two cases: in the third column the network is the original one, in the fourth column the architecture is enriched with the aforementioned noise layers. Moreover, we also create average HQ style and US clinical style in the feature space to be used as target for the training process.
To give some quantitative scores, we use FID between the generated set of images and the style images. For each image the index is calculated over some 299×299 patches and then averaged. Moreover, since the LQ to HQ task allows to compare against target HQ images, we provide also PSNR and SSIM scores.

	LQ to HQ	Seg to HQ	HQ to Clinical
Original Net	147.50	307.50	268.50
Noise Injection	135.25	319.25	250.25
Average Style	138.50	-	230.25

Table 4.7: FID for the different tasks

LQ to HQ	18.png		34.png		Set average	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
Original Net	55.5%	17.55	74.8%	27.29	71.4%	25.78
Noise Injection	52.3%	16.71	71.8%	25.20	68.4%	23.87
Average Style	56.4%	18.53	73.6%	26.20	71.7%	25.96

Table 4.8: LQ to HQ scores.

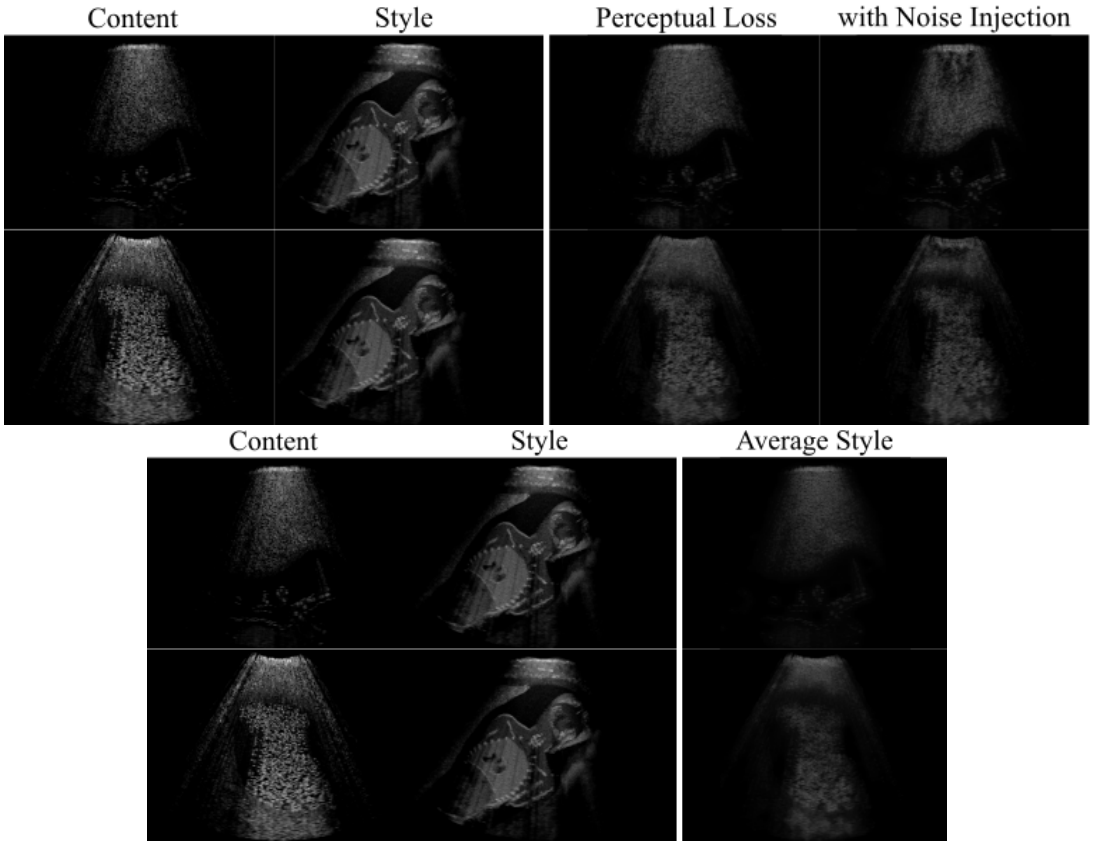

Figure 4.7: LQ to HQ images obtained with perceptual loss network.

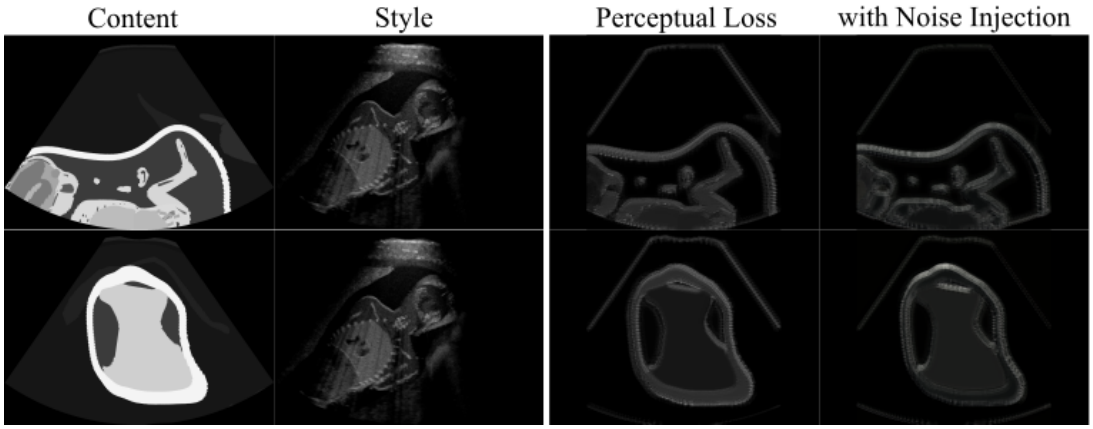

Figure 4.8: Segmentation to HQ images obtained with perceptual loss network.

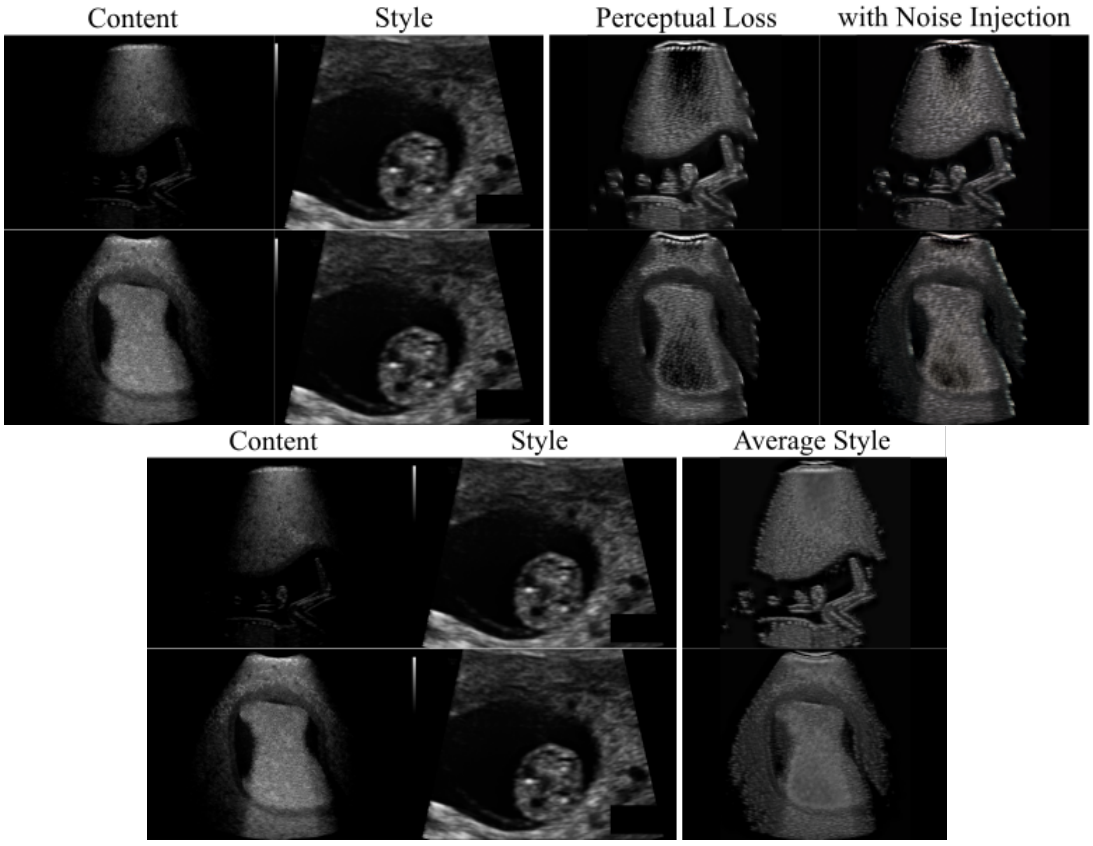

Figure 4.9: HQ to Clinical images obtained with perceptual loss network.

Chapter 5

Discussion

5.1 Low-quality to High-quality Style Transfer

Figure 4.2.b is a representation of the pure HQ style, which is interpreted as a fragmented patchwork of grayscale texture regions surrounded by dark areas. Indeed, the style image contains too many black regions, which are all accounted for in the overall style, and in general the images are very dark. Some figures may help too better understand the situation: in the reference style image, the 62% of the pixels are zeros, while the 72% has an intensity smaller than $10/256$ and only the 1.5% is brighter than $100/256$. This means that it might be difficult to extract a distinctive style, as in other contexts an artistic style may be (e.g Kandinskij paintings, a common example in literature that works very well). Our idea is that the HQ style transfer simply happens by surrounding the content with some texture and dark areas, and by reducing the overall brightness. Moreover, the loss plots in Figure 4.2, show a strange behavior of the optimization: when the content is weak, the style loss decreases by a factor of $1e4$, while when the content influence is strong, the style loss decreases only by a factor of 10 at most, being pretty high in the end ($1e6$).

In the augmented style loss approach (cf. Fig. 4.3), the segmentation map of the style image might help to mask out black regions, but there are some drawbacks with this solution. First of all, the style segmentation might not be available. Even when it is available, it might be over-informative, as it happens for our reference image (cf. Fig 3.1), meaning that it contains more "objects" than the one present in the low-quality/high-quality images. This produce an output that may be more correct style-wise, but the related scores are not trustworthy, since they compare two images that are effectively different. Anyway, this approach really helps in situations where the segmentation map is more consistent with the actual content, e.g. `18.png` (cf. Fig. 3.1) gives a SSIM of 57% with unpaired images. In general, one could also fine-tune the segmentation labels to obtain a more faithful result.

Figure 4.3 compares NST behaviour with paired and unpaired style images. Of course, the chosen style image strongly influences the outcome, but a difference of 13% in the SSIM cannot only be related to a 'good' or 'bad' style image, but it means that the style is not generalised very

well. While it is true that the style does not have any spatial information about its content, it is also true that if one applies the style on its own content the statistics should be more appropriate, leading to better outcomes. Instead, when the augmented style loss is used, the style is enriched with the spatial information from the segmentation masks: in the unpaired case it leads to the aforementioned issues; in the paired case it is welcome because it solves the problems (e.g. if the style in the head section is black, also in the outcome it will be black).
The average style (Fig 4.4.d) helps to better generalise the style because the gram matrix is calculated on the averaged statistics extracted from the style images. Indeed, the scores are higher and visually on the output image there is no "style-noise" outside the content.
The local NST (Fig 4.4.c) provides results that are similar to the NST with augmented style loss: the masking in the image space and the following localized optimization ensures that the content is well defined, but again the segmentation map strongly influences the outcome. With respect to the augmented style loss, this solution takes much longer to compute, but at least it can be obtained even if the style segmentation is not available.

The perceptual loss network seems to produce significantly better outputs than the NST optimization, with the great advantage of being real-time. Indeed, the SSIM is roughly 20% higher for the network output than for the optimization outcomes, and it is consistently over 70% for the tested images, with some exceptions on particular bad samples (e.g. image `18.png`).
Moreover, by using an average style calculated in the style feature space as target for the training, the SSIM slightly improves (on average).
The addition of the noise layers in the decoding section of the network does not provide any significant improvement in SSIM or PSNR, which instead are reduced. Indeed, visually the noise produce some dark textures in homogeneous areas (e.g. the top of the images), which may be the reason for this reduction in the scores. However, in the end the FID is smaller than with the other implementations, which is good.

5.2 Segmentation to High-quality Style Transfer

Regarding the segmentation to high-quality style transfer, the outputs are not satisfying in almost every case. The segmentation map is often different from the actual target. For example, in the reference it shows the head of the fetus and the placenta, while is missing the texture formed by the input rays when they cross the tissues. These differences are not accounted for by the scores, which consequently are substantially meaningless. Qualitatively, the augmented style loss reproduces the style texture more uniformly, but since the content and style are unpaired, regions that are supposed to be dark in the HQ image (e.g. head and back of the fetus) are instead filled.

The outputs of the perceptual loss network are still uniform as the inputs. Only around the edges one can notice the style texture, but the objects are still homogeneous. Unfortunately, the randomness introduced by the noise layers do not represent a noticeable improvement.

5.3 High-quality to Clinical Style Transfer

The clinical style is very similar to the HQ style from the optimization point-of-view, despite the fact that the style images are diverse. Indeed, it is visually clear that the clinical images are more realistic. However, the NST outcomes in figure 4.6 are not much different from the ones seen in Figure 4.4, probably the main differences being the coarser texture and the higher brightness of certain spots. Now the segmentation map of the style is not available, thus the augmented style loss masks the style in the feature space with the content masks, which may lead to different results depending on the actual style in the masked region, e.g. dark regions or texture regions. The average style in the feature space seems to help the generalization, but without any quantitative result it is quite hard to say. Qualitatively, the average style in the Gram space seems to reproduce more faithfully the brightness peaks which are present in the clinical style and gives a nicer representation of the tissues.

Once again, the perceptual loss network transfers the style better than the NST optimization: the stylized image is definitely more realistic than the input content. However, there still are some problems, indeed the edges are too sharp and tend to be brighter on the right side, and homogeneous areas tend to become black (especially on the top). The addition of noise layers reduces the extension of the dark regions, which is most welcome. This improvement is probably the cause for the better FID index. Eventually, using the average style helps the generalization, as proved by the correspondent FID.

Chapter 6

Conclusion

The results show that the perceptual loss network outperforms the Neural Style Transfer optimization. In particular, in the low-quality to high-quality task the network achieves a SSIM of 71%, compared to the 54.8% obtained with NST. Unfortunately, for the high-quality to clinical task it is hard to compare quantitatively the two algorithms, but visually we can say that the "stylization" provided by the learning approach is definitely more consistent with the input style.
The segmentation maps do not produce good result when used as contents, but they can help the style loss to gain some spatial information. Moreover, one can exploit the whole style dataset available: calculating an average style over the feature space leads to a better style generalization, which means a style transfer independent on the actual chosen style image.
Furthermore, the perceptual loss network runs in real-time on GPU and in a few seconds on CPU, which in any case is greatly faster than the NST optimization.
In general, the final results are not optimal, but for sure they are a good starting point for further research on the topic.

Bibliography

[1] A. Al Bahou, C. Tanner, and O. Goksel. Scatgan for reconstruction of ultrasound scatterers using generative adversarial networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1674–1677, 2019.

[2] H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Transactions on Image Processing*, 14(10):1647–1659, 2005.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

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

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[10] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer, 2017.

[11] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them, 2014.

[12] Oliver Mattausch, Maxim Makhinya, and Orcun Goksel. Realistic ultrasound simulation of complex surface models using interactive monte-carlo path tracing. *Computer Graphics Forum*, 37(1):202–213, 2018.

[13] Vitale Santiago, José Ignacio Orlando, Emmanual Iarussi, and Ignacio Larrabide. Improving realism in patient-specific abdominal ultrasound simulation using cyclegans, 2019.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[15] Thomas L. A. van den Heuvel, Dagmar de Bruijn, Chris L. de Korte, and Bram van Ginneken. Automated measurement of fetal head circumference using 2d ultrasound images, 2018.

[16] T.L.A. van den Heuvel, Dagmar de Bruijn, Chris L. de Korte, and Bram van Ginneken. Automated measurement of fetal head circumference, jul 2018.

[17] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.

[19] M. L. Østergaard, C. Ewertsen, L. Konge, E. Albrecht-Beste, and M. Bachmann Nielsen. Simulation-based abdominal ultrasound training - a systematic review, 2016.