Content Representation for Neural Style Transfer Algorithms based on Structural Similarity

Philip Meier, Volker Lohweg

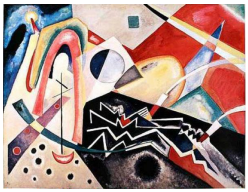
inIT – Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
Campusallee 6, 32657 Lemgo, Germany
{philip.meier,volker.lohweg}@th-owl.de

1 Introduction

Neural style transfer (NST) techniques allow to automatically merge the content of one image and the artistic style of another. This goal can be intuitively conveyed to a layperson with only three images (cf. Figure 1). Due to this simplicity, NST has gained some attraction for recreational use. In this context it is desirable to have a general purpose algorithm that is able to yield good results for a large variety of input images. If the quality of the new image is poor, the user might accept it anyway or simply move on without further consequences.

This changes drastically if NST is embedded into a professional environment, since it is usually not possible to swap out the inputs. For a single image and simple artistic styles a skilled artisan might be able to correct

(a)



(b)

(c)

Figure 1: Example of a NST by merging the content of (a) and style of (b) into a new image (c) with our proposed approach. Details about the utilised images and methods are presented in Section 4.

the results manually. While this partially defeats the goal to perform the stylisation automatically, it is also not feasible for more complex tasks. For example the stylisation of a video has to be performed for every frame. With a reasonable frame rate even short video clips comprise enough frames to render the manual stylisation practically impossible or at least infeasible. Even for single images the stylisation with a complex style such as the *intaglio* style [1] may be very time consuming. The transfer of the intaglio style to an image is currently performed manually which takes a skilled artisan around three months to complete.

It is thus crucial to a have a variety of alternative methods to choose from in order to increase the chances that at least one approach yields sufficient results. Within this contribution we propose a drop-in alternative to the way the content is represented within an NST algorithm; an area that so far has received little attention. First, however, we present the work related to our approach.

2 Related Work

Before presenting the related work in the following subsections, we introduce the notation we utilised throughout this contribution. Matrices are denoted by bold and not-italicised letters, e. g. a greyscale image of height H and width W could be denoted by $\mathbf{I} \in \mathbb{R}^{H \times W}$. Tensors with more than two dimensions are denoted by bold and not-italicised letters without serifs, e. g. an RGB image of the same spatial size as before with $C = 3$ channels could be denoted by $\mathsf{I} \in \mathbb{R}^{H \times W \times C}$. Furthermore, unless explicitly stated otherwise, all elementary arithmetic operations as well as assertions are performed elementwise. Finally, we utilise the sum symbol $\sum$ without indices to denote the sum of elements of the following matrix or tensor. In the same style, the average of a matrix or tensor, i. e. the elementwise sum divided by the number of elements, is denoted by $\overline{\sum}$.

2.1 Neural style transfer

The field of NST emerged in 2016 as a subfield of non-photorealistic rendering (NPR) pioneered by Gatys et al. [2]. NST algorithms allow to merge the content of one image $\mathbf{I}_\mathrm{C}$ with style of another $\mathbf{I}_\mathrm{S}$ into a new

image $\mathbf{I}$ (cf. Figure 1). Opposed to traditional NPR algorithms [3], NST methods do not operate on the pixel space or a handcrafted feature space, but rather on a learned feature space of a pre-trained convolutional neural network (CNN). This CNN is called *encoder* E, where $E^l(\mathbf{I})$ denotes the returned feature map of the image $\mathbf{I}$ by passing it through the CNN up to and including layer l.

In general, the synthesis is performed by optimising the image $\mathbf{I}$ in order to minimise a loss function $\mathcal{L}$:

$$\arg\min_{\mathbf{I}} \mathcal{L}(\mathbf{I}, \mathbf{I}_\mathrm{C}, \mathbf{I}_\mathrm{S}). \tag{1}$$

The loss function $\mathcal{L}$ is called *perceptual loss* and represents how well the synthesized image portrays the targeted content and style. In the most common formulation the perceptual loss $\mathcal{L}$ is a weighted sum of two loss functions $\mathcal{L}_\mathrm{C}$ and $\mathcal{L}_\mathrm{S}$, which represent the content and style agreement independently:

$$\mathcal{L}(\mathbf{I}, \mathbf{I}_\mathrm{C}, \mathbf{I}_\mathrm{S}) = \lambda_\mathrm{C} \cdot \mathcal{L}_\mathrm{C}(\mathbf{I}, \mathbf{I}_\mathrm{C}) \; + \; \lambda_\mathrm{S} \cdot \mathcal{L}_\mathrm{S}(\mathbf{I}, \mathbf{I}_\mathrm{S}). \tag{2}$$

The weights λ_C and λ_S are utilised to steer the synthesis towards a focus on content or style. Setting one of the weights to zero, the NST algorithm is transformed into a neural content reconstruction (NCR) or a style synthesis, respectively.

The content agreement is represented with the squared error (SE) between the feature maps of the image $\mathbf{I}$ and the content image $\mathbf{I}_\mathrm{C}$ on the layer l_C of the encoder E:

$$\mathcal{L}_{\mathrm{C,\,SE}}(\mathbf{I}, \mathbf{I}_\mathrm{C}) = \overline{\sum} \big(E^{l_\mathrm{C}}(\mathbf{I}) - E^{l_\mathrm{C}}(\mathbf{I}_\mathrm{C})\big)^2. \tag{3}$$

The feature maps on layer l_C are sparse representations of the objects or the content of the given image. For the sake of an expressive representation, it is crucial that the encoder is able to recognize the motifs within the images. Thus, it is common to utilise an encoder trained on large datasets to include a high degree of variety of motifs. Furthermore, the layer l_C is usually picked from deep within the CNN. Since the depth of the layer correlates with the abstraction of the recognised objects, a deeper layer ensures that the feature maps represent the actual content invariantly to irrelevant details [4].

Gatys et al. [5] suggest to treat the style of an image $\mathbf{I}$ as texture. Thus, NST is closely related to the field of texture synthesis. One

possible approach to represent the texture of a feature map is a stochastic one [6]. This assumes that if a given statistic of two feature maps match, the underlying texture and thus style is the same. In the original formulation the authors chose a correlation based approach. Given a set of column vectors arranged in a matrix $\mathbf{X}$, the matrix product, i. e. not the elementwise product, of its transposed version with itself is the Gram matrix [7]:

$$\text{gram} : \mathbb{R}^{N\times C} \to \mathbb{R}^{C\times C}, \quad \text{gram}(\mathbf{X}) = \mathbf{X}^T \cdot \mathbf{X}.$$

Depending on its position, each element within the Gram matrix is the inner product of the corresponding column vectors. Thus, the Gram matrix is an efficient way to calculate the correlation between all combinations of channels of a feature map. With this the style agreement for layer l_S is represented by the SE of the normalised Gram matrices:

$$\begin{aligned} \mathcal{L}_{\text{S},\,l_\text{S}}(\mathbf{I},\,\mathbf{I}_\text{S}) &= \overline{\sum}\big(g\big(E^{l_\text{S}}(\mathbf{I})\big) - g\big(E^{l_\text{S}}(\mathbf{I}_\text{S})\big)\big)^2 \\ \text{with} \quad g : \mathbb{R}^{H\times W\times C} &\to \mathbb{R}^{C\times C}, \quad g(\mathbf{X}) = \frac{\text{gram}(\text{spatvec}(\mathbf{X}))}{H\cdot W}. \end{aligned} \tag{4}$$

Here, spatvec$(\cdot)$ denotes a special case of the vectorisation function [7] that only collapses the spatial dimensions. Opposed to the content loss $\mathcal{L}_{\text{C},\,\text{SE}}$ in (3), the overall style loss $\mathcal{L}_\text{S}$ includes a weighted sum over multiple layers L_S in order to capture style elements of varying sizes and levels of detail:

$$\mathcal{L}_\text{S}(\mathbf{I},\,\mathbf{I}_\text{S}) = \sum_{l_\text{S}\in L_\text{S}} \lambda_{l_\text{S}} \cdot \mathcal{L}_{\text{S},\,l_\text{S}}.$$

In the few years since the emergence of NST a multitude of variants were developed. The following overview does not claim to be exhaustive. The authors refer the interested reader to [8] for a more complete review.

The NST approach explained above is usually dubbed *slow* NST, since the synthesis of a single image is computationally expensive. For each iteration step of (1) the synthesised image $\mathbf{I}$ has to be passed through the encoder E. To overcome this Johnson et al. [9] and Ulyanov et al. [10] independently proposed to train a *styliser* CNN S with a variant of the perceptual loss in (2):

$$\arg\min_S \mathcal{L}(\mathbf{I},\,\mathbf{I}_\text{C},\,\mathbf{I}_\text{S}) \quad \text{with} \quad \mathbf{I} = S(\mathbf{I}_\text{C}).$$

After the training phase, the styliser S performs stylisation of an image $\mathbf{I}$ in a single pass. These methods are thus called *fast* NST. Depending on the architecture, the styliser can generate single [9, 10, 11], multiple [12], or arbitrary styles [13]. Although the fast methods achieve remarkable results with respect to the computing cost, the slow or iterative method is still regarded as *gold standard* and is usually utilised as ground truth for comparison [8].

The field of NST suffers from a lack of objective methods to compare the results of different approaches. Usually the evaluation is performed only qualitatively by comparing few selected images [14, 15]. The validity of this approach is fairly low, since the achieved results might not be generalisable to arbitrary images.

Sanakoyeu et al. [11] presented professional art historians with sets of stylisation results generated by different algorithms and let them decide which method performed best. While this excludes the subjectiveness of the authors, due to the human judgement, albeit qualified, it is still not objective. Furthermore this measure is hardly reproducible, since even if other researchers had access to the same professionals, it is questionable if their judgement is invariant.

Sanakoyeu et al. [11] also introduced the *style transfer deception rate* to assess the quality. They trained a CNN to classify paintings according to their artists. The style transfer deception rate is defined as the fraction of stylised images that fool this CNN and are classified as originals from the respective artist. This is an objective and reproducible measure, but without further investigation it is unknown, if the classifier infers as expected.

We acknowledge that the objective comparison of NST algorithms is an open field of research, but we will not expand on it within this contribution. An objective quality assessment of unstylised images is presented in the following subsection.

This lack of objective comparison did not hinder the development of other NST variants. Many authors deal with alternative formulations of the style loss in (4). Other stochastic approaches include for example the channelwise histogram matching of the feature maps [15]. Li and Wand [16] proposed to represent the style with a structural approach [6] by matching spatial patches extracted out of the feature maps.

In contrast, to the best knowledge of the authors, no drop-in replacement for the content loss has been formulated. Sanakoyeu et al. propose a *style aware content loss* as part of a fast NST architecture [11]. It is still based on the SE of feature maps, but opposed to other methods the feature maps are extracted out of the trainable styliser S instead of a pretrained and fixed encoder E.

This contribution will push into this void and propose an alternative to the content loss in (3).

2.2 Structural similarity

The field of image quality assessment objectively measures the quality of images. For example, given a reference or ground truth image, methods aim to quantify the amount of distortion introduced by a reconstruction from a lossy compression or an NCR. Traditional methods include the SE or the peak-signal-to-noise ratio. Their meaning is however limited, since they align poorly with the human perception, i. e. multiple images that have the same SE towards the ground truth might have vastly different qualities to a human observer.

In order to overcome this, Wang et al. [17] introduced the structural similarity (SSIM) index. The SSIM index compares the luminance, contrast, and structure of two greyscale images $\mathbf{I}$ and $\mathbf{I}_\mathrm{R}$ independently of each other. The image $\mathbf{I}$ is a distorted version of the reference image $\mathbf{I}_\mathrm{R}$. The SSIM components may vary significantly throughout the images and thus the SSIM index is calculated locally with a window $\mathbf{w}$ normalised to unit sum $\sum \mathbf{w} = 1$. With this windowed approach the SSIM index is not a scalar measure, but rather a map for local intensity of the distortion. It is calculated as follows:

$$SSIM(\mathbf{I}, \mathbf{I}_\mathrm{R}) = l(\mathbf{I}, \mathbf{I}_\mathrm{R})^\alpha \cdot c(\mathbf{I}, \mathbf{I}_\mathrm{R})^\beta \cdot s(\mathbf{I}, \mathbf{I}_\mathrm{R})^\gamma. \tag{5}$$

Here l, c, and s denote the luminance, contrast, and structure component, respectively. The exponents α, β, and γ are used to adjust the weighting between the individual components. The components are calculated as

follows:

$$l(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = \frac{2 \cdot \mathbf{m}_\mathbf{I} \cdot \mathbf{m}_{\mathbf{I}_\mathrm{R}} + \epsilon_l}{\mathbf{m}_\mathbf{I}^2 + \mathbf{m}_{\mathbf{I}_\mathrm{R}}^2 + \epsilon_l}, \tag{6a}$$

$$c(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = \frac{2 \cdot \mathbf{s}_\mathbf{I} \cdot \mathbf{s}_{\mathbf{I}_\mathrm{R}} + \epsilon_c}{\mathbf{s}_\mathbf{I}^2 + \mathbf{s}_{\mathbf{I}_\mathrm{R}}^2 + \epsilon_c}, \tag{6b}$$

$$s(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = \frac{\mathbf{s}_{\mathbf{I},\,\mathbf{I}_\mathrm{R}} + \epsilon_s}{\mathbf{s}_\mathbf{I} \cdot \mathbf{s}_{\mathbf{I}_\mathrm{R}} + \epsilon_s}. \tag{6c}$$

Here $\mathbf{m}_\mathbf{I}$, $\mathbf{m}_{\mathbf{I}_\mathrm{R}}$, $\mathbf{s}_\mathbf{I}^2$, $\mathbf{s}_{\mathbf{I}_\mathrm{R}}^2$, and $\mathbf{s}_{\mathbf{I},\,\mathbf{I}_\mathrm{R}}$ are estimators for the local mean, variance, and covariance, respectively. They can be efficiently calculated by a discrete two-dimensional convolution with the window $\mathbf{w}$:

$$\begin{gathered}
\mathbf{m}_\mathbf{I} = \mathbf{I} * \mathbf{w}, \quad \mathbf{m}_{\mathbf{I}_\mathrm{R}} = \mathbf{I}_\mathrm{R} * \mathbf{w}, \\
\mathbf{s}_\mathbf{I}^2 = \mathbf{I}^2 * \mathbf{w} - \mathbf{m}_\mathbf{I}^2, \quad \mathbf{s}_{\mathbf{I}_\mathrm{R}}^2 = \mathbf{I}_\mathrm{R}^2 * \mathbf{w} - \mathbf{m}_{\mathbf{I}_\mathrm{R}}^2, \\
\mathbf{s}_{\mathbf{I},\,\mathbf{I}_\mathrm{R}} = (\mathbf{I} \cdot \mathbf{I}_\mathrm{R}) * \mathbf{w} - \mathbf{m}_\mathbf{I} \cdot \mathbf{m}_{\mathbf{I}_\mathrm{R}}.
\end{gathered}$$

In (6) ϵ_l, ϵ_c, and ϵ_s denote small positive constants. They are incorporated to avoid numerical instabilities if the denominators would be close or equal to zero without them. Wang et al. [17] recommend to calculate them with respect to a fixed small constant K and the dynamic range L, i. e. the maximum possible absolute value, of the input values:

$$\epsilon = (K \cdot L)^2. \tag{7}$$

The SSIM index is symmetric $SSIM(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = SSIM(\mathbf{I}_\mathrm{R},\, \mathbf{I})$, is bounded $-1 \leq SSIM(\mathbf{I}_\mathrm{R},\, \mathbf{I}) \leq 1$, and has a unique maximum for $\mathbf{I}_\mathrm{R} = \mathbf{I}$ at $SSIM(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = 1$, with respect to the input images $\mathbf{I}$, $\mathbf{I}_\mathrm{R} \geq 0$ [17, 18].

One common simplification of the SSIM is to weigh all components equally ($\alpha = \beta = \gamma = 1$) and set $\epsilon_1 = \epsilon_l$ as well as $\epsilon_2 = \epsilon_s = \epsilon_c/2$. This results in the simplified SSIM index:

$$sSSIM(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = S_1(\mathbf{I},\, \mathbf{I}_\mathrm{R}) \cdot S_2(\mathbf{I},\, \mathbf{I}_\mathrm{R}).$$

Under these assumptions the non-structural component S_1 is equal to the luminance component l and the contrast and structure component c and s are collapsed into a single structural component S_2:

$$S_1(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = \frac{2 \cdot \mathbf{m}_\mathbf{I} \cdot \mathbf{m}_{\mathbf{I}_\mathrm{R}} + \epsilon_1}{\mathbf{m}_\mathbf{I}^2 + \mathbf{m}_{\mathbf{I}_\mathrm{R}}^2 + \epsilon_1}, \quad S_2(\mathbf{I},\, \mathbf{I}_\mathrm{R}) = \frac{\mathbf{s}_{\mathbf{I},\,\mathbf{I}_\mathrm{R}} + \epsilon_2}{\mathbf{s}_\mathbf{I}^2 + \mathbf{s}_{\mathbf{I}_\mathrm{R}}^2 + \epsilon_2}.$$

Besides assessing the quality of a reconstructed image $\mathbf{I}$, the SSIM index can also be used to perform the reconstruction [18, 19]. In order to optimise the image quality, $SSIM(\mathbf{I}, \mathbf{I}_\mathrm{C})$ has to be maximised. Since optimisations are usually posed as minimisation problems, this is equivalent to minimising $1 - SSIM(\mathbf{I}, \mathbf{I}_\mathrm{C})$. Brunet achieved the best results by minimising the loss

$$\mathcal{L}(\mathbf{I}, \mathbf{I}_\mathrm{C}) = \overline{\sum}[(1 - S_1(\mathbf{I}, \mathbf{I}_\mathrm{R})) + (1 - S_2(\mathbf{I}, \mathbf{I}_\mathrm{R}))], \tag{8}$$

which is the average of the linear approximation of $1 - sSSIM(\mathbf{I}, \mathbf{I}_\mathrm{C})$ [18].

3 Approach

The last section explained that the SSIM index is better aligned with human quality perception of images than the SE. The CNNs utilised within NST algorithms are modelled after the human visual system, albeit grossly simplified. The approach we propose within this section is a content loss $\mathcal{L}_{\mathrm{C,\ SSIM}}$ that compares the feature maps $E(\mathbf{I})$ in terms of the SSIM.

Before we go into the details of the proposed approach, we need to prove that the basic properties of the SSIM (cf. Subsection 2.2) also hold for $\mathbf{I}, \mathbf{I}_\mathrm{R} \in \mathbb{R}$. This step is required, since in general and opposed to pixels, feature maps $E(\mathbf{I})$ might contain negative values.

Proposition 1. *The SSIM index is symmetric with respect to the inputs* $SSIM(\mathbf{I}, \mathbf{I}_R) = SSIM(\mathbf{I}_R, \mathbf{I})$ *for* $\mathbf{I}, \mathbf{I}_R \in \mathbb{R}$.

Proof. Since the symmetry of the components in (6) is independent of the extended input space, Brunets proof holds [18, p. 40]. □

Proposition 2. *The luminance* $l(\mathbf{I}, \mathbf{I}_R)$ *is bounded by* $-1 < l(\mathbf{I}, \mathbf{I}_R) \leq 1$ *for* $\mathbf{I}, \mathbf{I}_R \in \mathbb{R}$.

Proof. The following is a trivial extension of Brunets proof [18, p. 39] for the extended input space. The proposition is proven by contradiction for two cases:

Case 1. $l(\mathbf{I}, \mathbf{I}_\mathrm{R}) \overset{!}{>} 1$

$$
\begin{aligned}
& (\mathbf{m_I} - \mathbf{m_{I_R}})^2 \geq 0 && \Leftrightarrow \quad (\mathbf{m_I} - \mathbf{m_{I_R}})^2 + \epsilon_l \geq \epsilon_l \\
\Leftrightarrow \quad & \mathbf{m_I^2} + \mathbf{m_{I_R}^2} + \epsilon_l \geq 2 \cdot \mathbf{m_I} \cdot \mathbf{m_{I_R}} + \epsilon_l && \Leftrightarrow \quad \frac{2 \cdot \mathbf{m_I} \cdot \mathbf{m_{I_R}} + \epsilon_l}{\mathbf{m_I^2} + \mathbf{m_{I_R}^2} + \epsilon_l} \leq 1 \\
\Leftrightarrow \quad & l(\mathbf{I}, \mathbf{I}_\mathrm{R}) \leq 1
\end{aligned}
$$

Case 2. $l(\mathbf{I}, \mathbf{I}_\mathrm{R}) \overset{!}{\leq} -1$

$$
\begin{aligned}
& (\mathbf{m_I} + \mathbf{m_{I_R}})^2 > -2\epsilon_l && \Leftrightarrow \quad (\mathbf{m_I} - \mathbf{m_{I_R}})^2 + \epsilon_l > -\epsilon_l \\
\Leftrightarrow \quad & \mathbf{m_I^2} + \mathbf{m_{I_R}^2} + \epsilon_l > -2 \cdot \mathbf{m_I} \cdot \mathbf{m_{I_R}} - \epsilon_l && \Leftrightarrow \quad \frac{2 \cdot \mathbf{m_I} \cdot \mathbf{m_{I_R}} + \epsilon_l}{\mathbf{m_I^2} + \mathbf{m_{I_R}^2} + \epsilon_l} > -1 \\
\Leftrightarrow \quad & l(\mathbf{I}, \mathbf{I}_\mathrm{R}) > -1
\end{aligned}
$$

□

Proposition 3. *The contrast and the structure components* $c(\mathbf{I}, \mathbf{I}_R)$ *and* $s(\mathbf{I}, \mathbf{I}_R)$ *are bounded by* $0 \leq c(\mathbf{I}, \mathbf{I}_R) \leq 1$ *and* $-1 \leq s(\mathbf{I}, \mathbf{I}_R) \leq 1$ *for* $\mathbf{I}, \mathbf{I}_R \in \mathbb{R}$.

Proof. Since the extended input space does not change the value range of estimated standard deviations $\mathbf{s_I}, \mathbf{s_{I_R}} \geq 0$ or the estimated covariance $\mathbf{s_{I,I_R}} \in [-1, 1]$, BRUNETs proof [18, pp. 39-40] holds. □

Corollary 1. *The SSIM index is bounded by* $-1 \leq SSIM(\mathbf{I}, \mathbf{I}_R) \leq 1$ *for* $\mathbf{I}, \mathbf{I}_R \in \mathbb{R}$.

Proof. SSIM index is defined as the product of the individual components in (5). By combining Proposition 2 and Proposition 3, we arrive at the desired conclusion. □

Proposition 4. *The SSIM index has a unique maximum* $SSIM(\mathbf{I}, \mathbf{I}_R) = 1$ *at* $\mathbf{I} = \mathbf{I}_R$ *for* $\mathbf{I}, \mathbf{I}_R \in \mathbb{R}$.

Proof. The proof is structured into two parts: First, we prove that $SSIM(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$ if $\mathbf{I} = \mathbf{I}_\mathrm{R}$. Subsequently, we prove that $SSIM(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$ is unique for $\mathbf{I} = \mathbf{I}_\mathrm{R}$.

1. By substituting $\mathbf{I} = \mathbf{I}_\mathrm{R}$ in (5), we arrive at the desired conclusion.

2. Brunet proved that $l(\mathbf{I}, \mathbf{I}_\mathrm{R}) = c(\mathbf{I}, \mathbf{I}_\mathrm{R}) = s(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$ and subsequently $SSIM(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$ is uniquely obtained by $\mathbf{I} = \mathbf{I}_\mathrm{R}$ independent of the input space. The only other way to obtain $SSIM(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$ through a product of the components would be by $l(\mathbf{I}, \mathbf{I}_\mathrm{R}) \overset{!}{=} -1$, $c(\mathbf{I}, \mathbf{I}_\mathrm{R}) = 1$, and $s(\mathbf{I}, \mathbf{I}_\mathrm{R}) = -1$. However, this combination is invalid, since after Proposition 2 $l(\mathbf{I}, \mathbf{I}_\mathrm{R}) \overset{!}{=} -1$ is not achievable.

□

In summary the SSIM index retains its symmetry, boundedness, and its unique maximum for real-valued inputs. Naturally these properties also apply to the derived simplified SSIM.

Based on (8) we propose

$$\mathcal{L}_{\mathrm{C},\,\mathrm{SSIM}}(\mathbf{I}, \mathbf{I}_\mathrm{C}) = \lambda_1 \cdot \mathcal{L}_{\mathrm{C},\,S_1}(\mathbf{I}, \mathbf{I}_\mathrm{C}) + \lambda_2 \cdot \mathcal{L}_{\mathrm{C},\,S_2}(\mathbf{I}, \mathbf{I}_\mathrm{C})$$

with

$$\mathcal{L}_{\mathrm{C},\,S_1}(\mathbf{I}, \mathbf{I}_\mathrm{C}) = \overline{\sum}\big(1 - S_1\big(E^{l_\mathrm{C}}(\mathbf{I}),\, E^{l_\mathrm{C}}(\mathbf{I}_\mathrm{C})\big)\big),$$
$$\mathcal{L}_{\mathrm{C},\,S_2}(\mathbf{I}, \mathbf{I}_\mathrm{C}) = \overline{\sum}\big(1 - S_2\big(E^{l_\mathrm{C}}(\mathbf{I}),\, E^{l_\mathrm{C}}(\mathbf{I}_\mathrm{C})\big)\big).$$

as a drop-in replacement for the content loss in (3). Opposed to the traditional approach we do not compare every element pair of the feature maps individually, but rather their SSIM in the local neighbourhood of the window $\mathbf{w}$. The weights λ_1 and λ_2 are added to enable an emphasis on the non-structural or structural component, respectively.

We treat each channel pair of the feature maps $E^{l_\mathrm{C}}(\mathbf{I})$ and $E^{l_\mathrm{C}}(\mathbf{I}_\mathrm{C})$ independently, i. e. we calculate an SSIM map for each channel. Since the dynamic range might vary significantly between the channels, the stability constants ϵ_1 and ϵ_2 are also calculated channelwise according to (7). This calculation is performed apriori with the feature maps $E^{l_\mathrm{C}}(\mathbf{I}_\mathrm{C})$ of the content image. The other hyperparameters such as the component weights λ_1 and λ_2 as well as the utilised window $\mathbf{w}$ are determined empirically in the following section.

4 Evaluation

This section is divided into three subsections. At first, we describe the methods we utilised throughout this section[1]. Secondly, an objective evaluation of the proposed content loss $\mathcal{L}_{\mathrm{C,\ SSIM}}$ compared to the traditional formulation $\mathcal{L}_{\mathrm{C,\ SE}}$ is carried out. Lastly, a qualitatively comparison is performed with the content losses as part of an NST.

4.1 Methods

NCR and NST should be ideally feasible for arbitrary content images given that the motifs are known to the encoder and it thus is able to generate meaningful encodings. For the evaluation we utilise the *NPRGeneral* dataset as content images which include wide range of features [20]. The motifs comprise humans, objects, and landscapes. We center-cropped all images to 1024×640 pixels, which is the size of the smallest image in the dataset. An overview of the dataset is depicted in Figure 5 in the appendix. As style images we utilised *White Zig Zags* by KANDINSKY and *Landscape at Saint-Rémy* by VAN GOGH (cf. Figure 6 in the appendix). The former comprises large abstract forms as style elements while the latter is characterized by fine brush strokes. Both images are part of unnamed style image dataset proposed in [8].

For the image synthesis we mostly follow the procedure proposed in [21] which achieved good results and makes our results comparable. We utilise *VGG19* CNN [22] as encoder E. The CNN was trained on the *ILSVRC 2012* dataset [23] which is suitable for the motifs contained in the NPRGeneral dataset. Although the weights were trained with the *Caffe* framework [24], we implement our approach with the *PyTorch* framework [25]. We use $l_{\mathrm{C}} = \texttt{relu\_4\_2}$ and $l_{\mathrm{S}} \in L_{\mathrm{S}} = \{\texttt{relu\_1\_1}, \texttt{relu\_2\_1}, \texttt{relu\_3\_1}, \texttt{relu\_4\_1}, \texttt{relu\_5\_1}\}$ as layers for content and style feature maps, respectively. The style weights $\lambda_{l_{\mathrm{S}}}$ are calculated by $\lambda_{l_{\mathrm{S}}} = 1/n_{l_{\mathrm{S}}}^2$, where $n_{l_{\mathrm{S}}}$ denotes the number of channels of the feature map on layer l_{S}.

We also utilise an image pyramid with two levels to enhance the synthesis results. For the first level we stick to the procedure of [21] and resize the images to 800×500 pixels. On the second and final level we only

[1] We publish the source code to reproduce our results under `https://github.com/pmeier/GMA_CI_2019_ssim_content_loss`

increase the image size of the synthesized image $\mathbf{I}$ to 1024×640 pixels. The reference images are used in their unaltered form to avoid possibly introducing resampling artifacts in the ground truth. The resizing is performed with bilinear resampling. The synthesis is carried out by the L-BFGS optimisation algorithm [26] with 500 steps on the first and 200 steps on the second level as proposed in [21].

4.2 Neural Content Reconstruction

While NST algorithms lack a method to assess or compare their performance objectively, this is possible for NCR algorithms. Since the target of the reconstruction $\mathbf{I}_\mathrm{C}$ is known apriori, we define the mean SSIM of the reconstructed image $\mathbf{I}$ and the target image $\mathbf{I}_\mathrm{R} = \mathbf{I}_\mathrm{C}$ as *SSIM score*. Both images are priorly converted to greyscale with standard channel weights [27].

The reconstruction is started from a white noise image to avoid any bias from pre-existing content. Furthermore we repeat every reconstruction with five different random seeds and report the median result in order to reduce the influence of random effects. We explicitly state if we deviate from this scheme.

We set $\lambda_{\mathrm{C,\,SE}} = 10^{-3}$ and $\lambda_{\mathrm{C,\,SSIM}} = 10^{3}$ to achieve approximately the same overall magnitude of the content loss $\mathcal{L}_\mathrm{C}$. The stability constants are calculated according to (7) with $K_1 = 10^{-2}$ and $K_2 = 3 \cdot 10^{-2}$ as suggested in [17]. If not stated otherwise we use $\lambda_1 = 0.1$ and $\lambda_2 = 0.9$ as component weights and a 3×3 GAUSSIAN with standard deviation $\sigma = 1/3$ as window $\mathbf{w}$. The determination of these parameters is explained in the following experiments.

In a first experiment we benchmarked our SSIM based content loss $\mathcal{L}_{\mathrm{C,\,SSIM}}$ with different component weights λ_1 and λ_2 against SE based $\mathcal{L}_{\mathrm{C,\,SE}}$ with the NPRGeneral dataset. The SSIM component weights λ_1 and λ_1 were chosen, such that $\rho = \lambda_2/\lambda_1$ and $\lambda_1 + \lambda_2 = 1$. Table 1 depicts the results for selected images, while the results for the complete dataset are displayed in Table 2 in the appendix. In general, we observe that for our approach a higher structural weight λ_2 leads to an improved performance. The SSIM score with only the structural component $\rho \to \infty$ is similar compared to the traditional approach, while the better performing approach is dependent on the image. For the images *mac*, *oparara*, and *arch* the SSIM score drops off for higher structural weights

Table 1: Median SSIM score for an NCR of selected images in the NPRGeneral dataset with different loss functions.

image	$\mathcal{L}_{\text{C, SE}}$	$\mathcal{L}_{\text{C, SSIM}}$			
		$\rho = 0$	$\rho = 3$	$\rho = 9$	$\rho \to \infty$
arch	**0.14**	0.11	0.12	0.13	0.12
athletes	0.52	0.42	0.48	0.50	**0.53**
berries	0.33	0.26	0.31	0.33	**0.34**
mac	**0.48**	0.37	0.46	0.47	0.45
snow	**0.24**	0.17	0.21	0.00	0.00

λ_2. This is especially true for the images *snow* and *daisy*: since they comprise large homogenous areas with little structure, our approach is not able to synthesise any content for a high structure weight λ_2 in the majority of runs. The other extreme, i. e. non-structural only $\rho = 0$, is stable, but performs significantly worse than the traditional approach. This implies that a good balance between the component weights has to be found on a per-image basis. For the following experiments we chose $\lambda_1 = 0.1$ and $\lambda_2 = 0.9$ ($\rho = 9$), since it performed reasonably well for most images within NPRGeneral dataset. Furthermore, in order to reduce the evaluation space, we chose the image *berries* as proxy, because both approaches performed very similarly on it. Additionally, its SSIM scores are approximately in the middle between the best and worst reconstructible images *athletes* and *arch*, respectively.

Ideally, the former benchmark should be carried out on steady state behaviour, i. e. further optimisation steps should have negligible effects. Figure 2 depicts that a steady state of the SSIM score is not achieved within the usual range of $\leq 10^3$ steps. Both approaches only plateau after approximately 10^5 steps. Continuing the NCR at this point decreases the SSIM score again. This effect results from unconstrained optimisation: every pixel that gets fit by the optimisation outside of the closed interval [0, 1] becomes salt and pepper noise after clipping and thus reduces the quality. Our approach is stronger affected by this, since $\mathcal{L}_{\text{C, SSIM}}$ does not stagnate opposed to $\mathcal{L}_{\text{C, SE}}$.

Lastly, the window $\mathbf{w}$ has to be determined. Figure 3 depicts the SSIM scores for a box or Gauss window for varying radii. In general, the performance degenerates for a larger radius r. This is a result of the

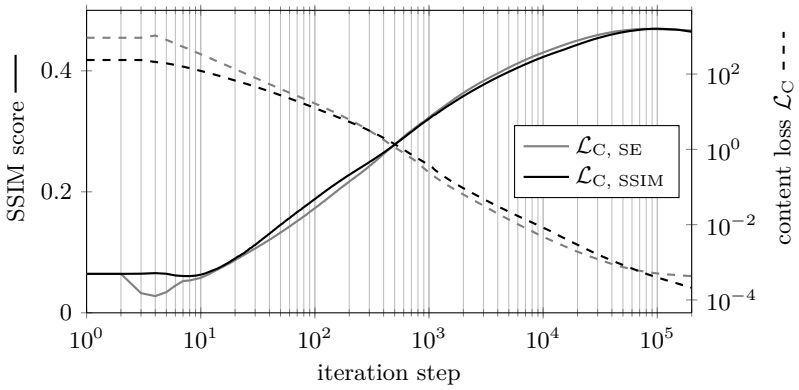


Figure 2: SSIM score and content loss $\mathcal{L}_C$ over the course of an optimisation. The data was recorded in a single run and without utilising the image pyramid.

significantly increased *receptive field* of $84 + 16 \cdot r$ pixel of the window $\mathbf{w}$ on the layer $l_C =$ `relu_4_2`. Within this contribution the receptive field is defined as the largest object in pixels within an image that can completely fit into a local neighbourhood of sequential convolutions or similar operations. The factor 16 in the receptive field calculation is formed by the three prior pooling operations within the encoder E with a stride of 2 and the fact that radius counts twice for the size of the window $\mathbf{w}$. Since larger receptive fields correspond to larger recognisable objects, they tend to suppress finer details, which in turn lowers the performance. Furthermore, Figure 3 shows that padding the feature maps prior to applying the window $\mathbf{w}$ is beneficial. If this step is omitted, the area near the edges is under-represented, which leads to a performance loss. This effect is amplified for greater radii. All in all, we chose a Gauss window with a radius $r = 1$ and padding for the further experiments.

In conclusion, we observed that no approach is objectively better than the respective other one within the carried out experiments. In general, our approach performs best with an emphasised structural component and small windows. After this quantitative evaluation we now compare our approach and the traditional qualitatively by incorporating them in an NST.

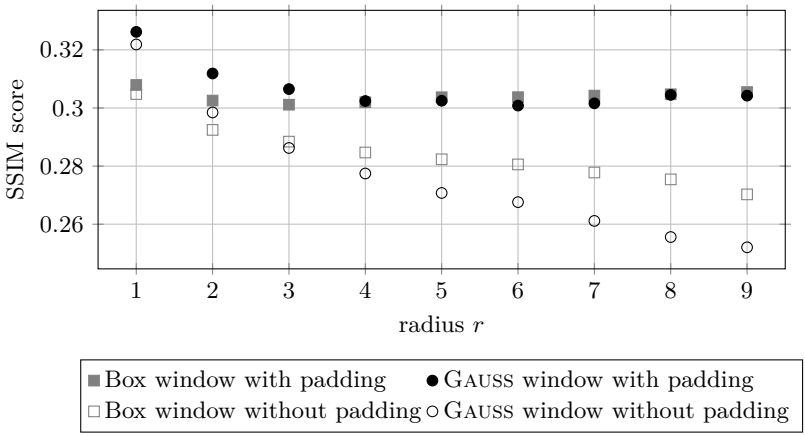


Figure 3: Median SSIM score of the NCR with the image *berries* over different window radii. The window $\mathbf{w}$ is square with a side length of $2r + 1$ pixels. The standard deviation σ of the Gaussian is set to $\sigma = r/3$.

4.3 Neural Style Transfer

The synthesis of the stylised image is started from the content image $\mathbf{I}_\mathrm{C}$, since the result converges faster to a visually appealing result. We set the style weight to $\lambda_\mathrm{S} = 1$, which leads to the suggested weight ratio $\lambda_\mathrm{C}/\lambda_\mathrm{S} = 10^{-3}$ from [21] for the traditional approach. The style images are also resized with the image pyramid, but are kept in their original aspect ratio.

We performed an NST with all combinations of content images $\mathbf{I}_\mathrm{C}$ from the NPRGeneral dataset and style images $\mathbf{I}_\mathrm{S}$. Within this subsection we only present parts of this experiment, but the interested reader can find all results in the accompanying repository.

The image *White Zig Zags* by Kandinsky (cf. Figure 6 in teh appendix) features large style elements which are separated by sharp edges. With this style both approaches yield virtually indistinguishable results (cf. Figure 7 in the appendix). *A Landscape at Saint-Rémy* by van Gogh on the other hand comprises fine brush strokes with mostly low contrast between them. In the synthesised image, our approach favours high contrast brush strokes (cf. Figure 4): due to the higher structural emphasis of the content loss, a high contrast style element minimises both the

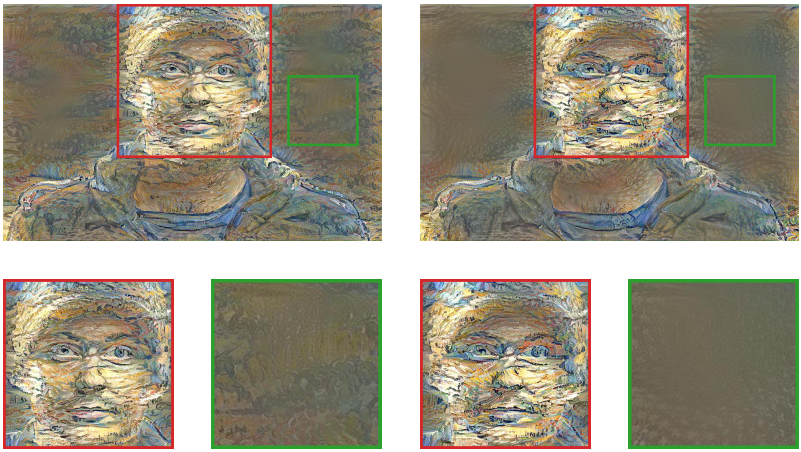


Figure 4: Results of the NST of the content image *rim lighting* and style image *Landscape at Saint-Rémy*. The left column was synthesised with the traditional and the right column with our proposed approach.

content and style loss. The resulting sharp edges are partially dominant enough to distort the content. This is especially visible for human faces, since the human mind is trained to subconsciously notice even slight differences. Although they are not part of this initial evaluation, styles which feature high contrast elements could benefit from our approach. Although both approaches struggle to stylise the homogenous areas in Figure 4, the traditional approach copes better with it. This is again a result of structure emphasis of our approach within an area without any structure.

5 Conclusion and Outlook

In this contribution we introduced a novel content representation as a drop-in replacement for neural style transfer algorithms. Opposed to the traditional approach it does not rate the content agreement on the squared error of the feature maps, but rather on their structural similarity. In a quantitative comparison, we found that our approach performs similarly to the traditional approach, if utilised within a neural content reconstruction. The better performing approach is determined on

a per-image basis, while our approach exhibits some instability for several images depending on the chosen weighting factors. Furthermore, as part of a neural style transfer our approach tends to emphasise high contrast style elements. While this leads to subjectively worse performance for the styles we utilised in our experiments, it might be an advantage for other styles.

Future work can expand on ours by investigating if the results of the qualitative evaluation hold in a more comprehensive study, i. e. performing the neural style transfer with a larger variety of styles. This should include styles with small, high-contrast elements, e. g. the intaglio style. Additionally, the investigation should evaluate the effect of our proposed content loss in the presence of other style losses.

References

[1] Rudolph L. van Renesse. *Optical Document Security.* Artech House, 3. edition, 2005.

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2001.

[4] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *Computing Research Repository (CoRR)*, 1311, 2013.

[5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS) 28*, 2015.

[6] Dongxiao Zhou. *Texture analysis and synthesis using a generic Markov-Gibbs image model.* PhD thesis, University of Auckland, 2006.

[7] Leslie Hogben, editor. *Handbook of Linear Algebra*. Chapman & Hall / CRC, 1. edition, 2007.

[8] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song. Neural style transfer: A review. *Computing Research Repository (CoRR)*, 1705, 2017.

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the 14^{th} European Conference on Computer Vision (ECCV)*, 2016.

[10] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *Computing Research Repository (CoRR)*, 1603, 2016.

[11] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the 15^{th} European Conference on Computer Vision (ECCV)*, 2018.

[12] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *Preprint in Computing Research Repository (CoRR)*, 1610, 2016.

[13] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *Computing Research Repository (CoRR)*, 1703, 2017.

[14] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of the 14^{th} European Conference on Computer Vision (ECCV)*, 2016.

[15] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *Computing Research Repository (CoRR)*, 1701, 2017.

[16] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.

[18] Dominique Brunet. *A Study of the Structural Similarity Image Quality Measure with Applications to Image Processing.* PhD thesis, University of Waterloo, 2012.

[19] Dominique Brunet, Sumohana S. Channappayya, Zhou Wang, Edward R. Vrscay, and Alan C. Bovik. *Optimizing Image Quality.* Springer International Publishing, 2018.

[20] David Mould and Paul L. Rosin. A benchmark image set for evaluating stylization. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling (Expressive) and Non-Photorealistic Animation and Rendering (NPAR)*, 2016.

[21] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. *Computing Research Repository (CoRR)*, 1611, 2017.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository (CoRR)*, 2014.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

[24] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *Computing Research Repository (CoRR)*, 1408, 2014.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Proceedings of the NIPS Autodiff Workshop*, 2017.

[26] Joel Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer-Verlag New York, 2. edition, 2006.

[27] International Telecommunication Union (ITU). Recommendation ITU-R BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, 2011.

Appendix

Table 2: Extension of Table 1. Median SSIM score for an NCR of the NPRGeneral dataset with different loss functions.

image	$\mathcal{L}_{\text{C, SE}}$	$\mathcal{L}_{\text{C, SSIM}}$			
		$\rho = 0$	$\rho = 3$	$\rho = 9$	$\rho \to \infty$
angel	**0.29**	0.20	0.25	0.27	**0.29**
arch	**0.14**	0.11	0.12	0.13	0.12
athletes	0.52	0.42	0.48	0.50	**0.53**
barn	**0.27**	0.21	0.26	**0.27**	**0.27**
berries	0.33	0.26	0.31	0.33	**0.34**
cabbage	0.41	0.32	0.39	0.41	**0.43**
cat	0.24	0.20	0.23	0.24	**0.25**
city	**0.33**	0.22	0.29	0.31	0.31
daisy	**0.50**	0.37	0.45	0.46	0.00
dark woods	**0.16**	0.10	0.14	0.15	0.15
desert	**0.33**	0.27	0.30	0.31	0.31
headlight	**0.49**	0.27	0.39	0.43	0.47
mac	**0.48**	0.37	0.46	0.47	0.45
mountains	**0.45**	0.32	0.39	0.41	0.44
oparara	**0.19**	0.12	0.16	0.18	0.17
rim lighting	**0.15**	0.09	0.12	0.13	0.13
snow	**0.24**	0.17	0.21	0.00	0.00
tomatoes	**0.42**	0.25	0.35	0.39	0.41
toque	**0.40**	0.30	0.37	0.38	0.39
yemeni	**0.40**	0.26	0.32	0.35	0.37

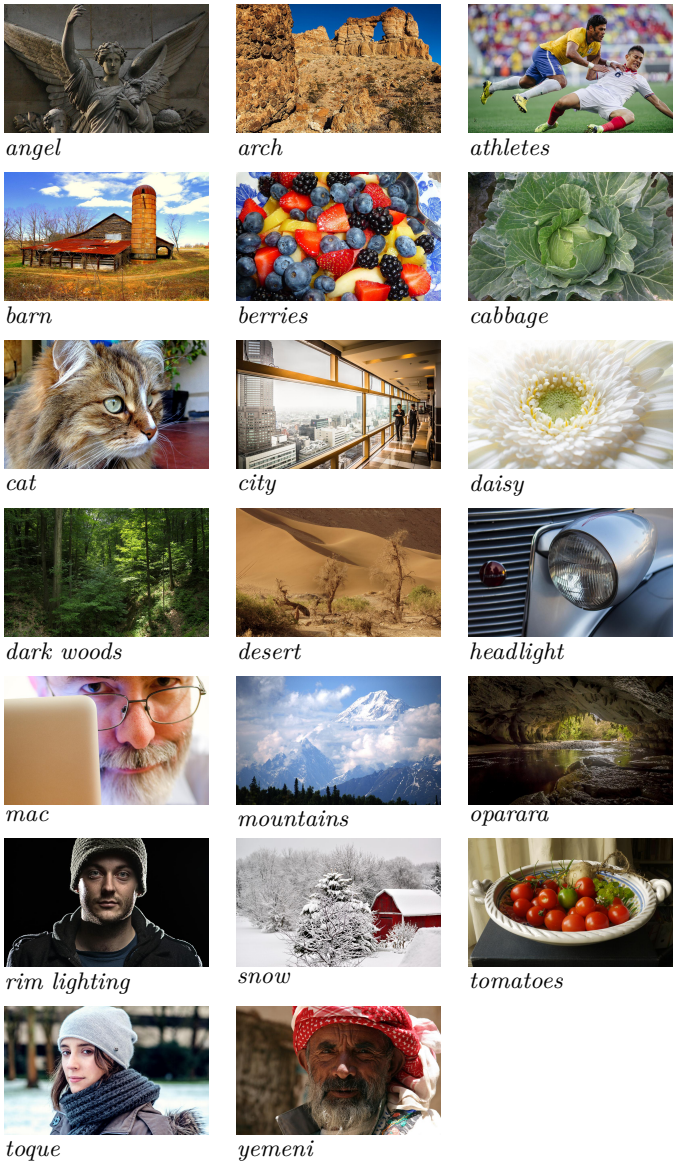


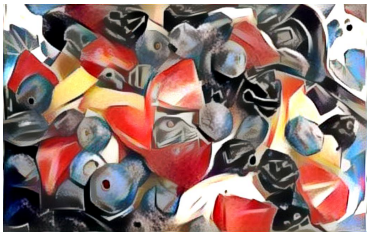
Figure 5: Overview over the images in the NPRgeneral dataset [20], which are utilised as content images.

(a) *White Zig Zags* by KANDINSKY

(b) *Landscape at Saint-Rémy* by VAN GOGH

Figure 6: Overview over the utilised style images.

(a)



(b)

Figure 7: Results of the NST of the content image *berries* and style image *White Zig Zags* with the traditional (a) and our proposed (b) content representation. Without further processing the differences are imperceptible.