TEXTURE UPSAMPLING AND ENHANCEMENT USING NEURAL SYNTHESIS

ROEMER HUGO WILLEM BLOM

university of groningen

Bachelor's Thesis

December 2023

SUPERVISORS:
Dr. C. Tursun
Prof. J. Kosinka

ABSTRACT

Visual texture of surfaces plays an important role in human visual perception, particularly in how we discern and interpret material properties and spatial relationships in our environment. Texture synthesis is the process of algorithmically constructing a novel image texture from a sample, maintaining the visual appearance and essential characteristics of the sample. Gatys et al. (2015) developed a technique to synthesise textures using a model based on correlations in the feature space of a pre-trained convolutional neural network, namely VGG-19. Our research expands upon this foundation by exploring the method's adaptability to a different CNN architecture, demonstrating its efficacy with ResNet34. We also introduce a gradient threshold as a stopping criterion for the synthesis process, significantly enhancing computational efficiency without compromising texture quality. Further, we investigate various seed image types, especially in terms of their frequency domain content, to determine their impact on the synthesis outcome. Additionally, our study examines the relationship between scale invariance in textures and synthesis quality, utilising wavelet transform for a multi-scale analysis.

CONTENTS

LIST OF FIGURES

INTRODUCTION

1.1 DEFINITION

It is imperative for the advancement of our discussion to first and foremost define what exactly a texture is. When we think of textures, we will typically say that they are the "look" and "feel" of an object. Texture can thus refer to the visual appearance of the surface of an object or to the way in which its surface relief results in different touch sensations. This thesis will only be concerned with the visual aspect of texture, i.e. image textures. Furthermore, usually object surfaces are three dimensional, however, we will only be working with two dimensional representations. This boils down to visual two dimensional textures represented as RGB images. So from here on out, this is what we will refer to when we use the word "texture".

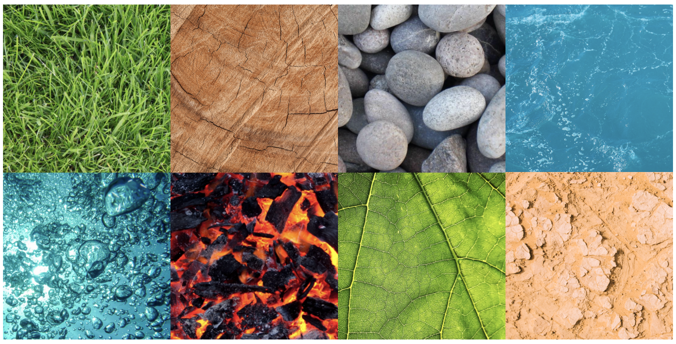


Figure 1: Examples of natural textures

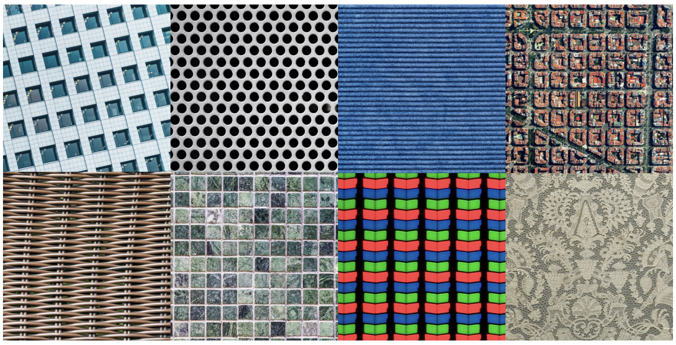


Figure 2: Examples of artificial textures

Textures can come in many different varieties. We observe textures on natural objects such as those displayed in Figure 1 as well as on artificial objects in Figure 2. Furthermore, textures can be ordered along a spectrum going from regular to stochastic. On one end, there exist regular textures, which follow well defined repetitive spatial patterns. On the other end, there are stochastic textures, which lack a repetitive pattern and are best described by a probabilistic model. There seems to be no structure or pattern to these textures. Also, even within these categories textures can differ greatly due to variations in the colour, shape, rotation, arrangement or density of their elementary parts. Due to this property of texture, it is difficult to give a concise definition that captures all dimensions of the concept of texture.

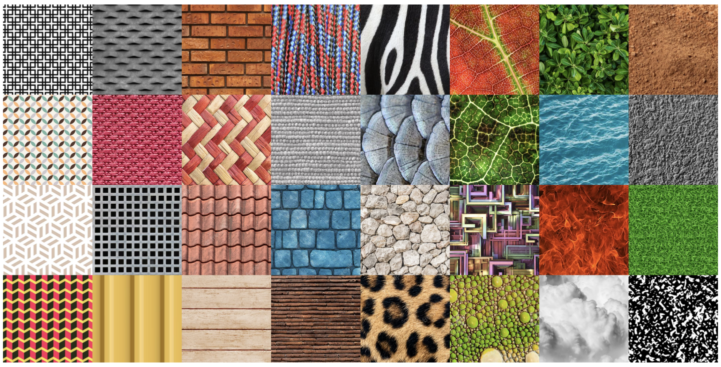


Figure 3: Decreasing texture regularity from left to right

Several researchers have attempted to formulate precise definitions of texture. Haralick [15] defines texture as an organised area phenomenon. This definition encapsulates the idea that textures consist of structural elements displaying a certain pattern across a surface. 'Primitives,' as he calls them, have specific spatial distributions, indicating that the arrangement and spacing of these elements are key characteristics in defining the texture. Cross and Jain [7] describe texture as a stochastic and occasionally periodic two-dimensional image field. This definition highlights the balance between randomness and periodicity in textures. The stochastic nature suggests variability in texture elements, while the periodic aspect acknowledges the presence of repeating patterns in many textures.

Portilla and Simoncelli, whose work will be further discussed in subsequent sections, have also contributed their definition of texture. They characterise textures as spatially homogeneous regions, marked by repetition with an element of randomisation. This definition acknowledges the uniformity in textures, while also recognising the inherent variability in elements such as location, size, colour, and orientation. This perspective emphasises the coexistence of order and randomness within textural patterns.

We can detect an apparent agreement regarding the significance of spatial homogeneity as a fundamental characteristic of textures. From a statistical perspective, homogeneity refers to the concept of statistical stationarity, implying that specific signal statistics within each texture region maintain consistent values. This particular attribute directly correlates with self-similarity, whereby patterns observed at various scales, while not completely identical, exhibit similar signal statistics [36].

1.2 texture perception

Our capacity to perceive texture plays a crucial role in our ability to understand scenes. Textures give important visual cues regarding depth, orientation, etc. The entire visual system is a complex process and it has been shown that texture perception is one of its foundational features [3, 24].

1.3 julesz' conjecture

Julesz was one of the first to systematically study human texture perception [24, 25]. Central to his research was establishing a statistical basis for human texture discrimination. He created images wherein he combined certain textures (see Figure 4). These textures were generated using stochastic processes, which are specified by their Nth order joint probability distribution. His reasoning behind these stimuli was that because they are devoid of any familiar cues, they deprive subjects of habitual recognition and force them to rely on more primitive mechanisms.

Julesz showed these images to participants for very short time periods and had them try to pre-attentively discriminate between textures. He found that certain lower-order statistics of textures could define our inability to discriminate between them. In specific, he demonstrated that many textures sharing the same second-order statistics could not easily be discriminated pre-attentively. This statistical approach to analysing texture served as the basis for many subsequent texture models.

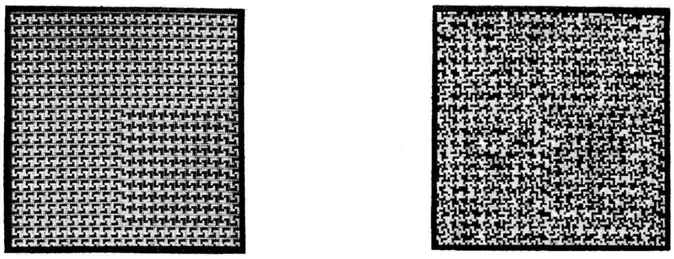


Fig. 9—Two fields which are each others complements. Discrimination is a result of different line structure seen.

Fig. 10—Identical with Fig. 9 but every fifth gray row and column is changed to black and white random dots.

Figure 4: Images used for research by Julesz

1.4 statistical analysis of textures

Julesz's research regarding the statistical classification of textures based on human perception prompted many others to build statistical models of textures. His first conjecture regarding second-order statistics of textures inspired a class of algorithms called the *random phase methods* [38], which utilise the square Fourier modulus equalling spatial auto-correlation, a second-order statistic.

Later is was found that textures could share second- and even third-order statistics while being perceptually distinct [5]. This finally led to Julesz proposing the concept of *textons* [26], which are discernible features like lines and corners. He conjectured that the first-order statistics of these textons are relevant for texture perception. Texton theory proposes the main axiom that texture perception is invariant to random shifts of the textons [23]. This axiom is used, for example, in the *stochastic dead leaves model* [4].

Largely inspired by the work of Julesz, statistical texture models try to capture the perceptual qualities of a texture in a summary statistic: a mathematical representation of what perceptually defines a texture.

1.5 texture models

Analysis of texture can also be reversed by means of synthesis. Exemplar-based synthesis employs the statistical analysis of texture to steer synthesis of novel texture images. It functions by first obtaining a set of statistics that can capture perceptual qualities of the example texture. Synthesis is then performed by enforcing these statistics onto some random seed image using an iterative optimisation scheme. In matching up the defined statistics, the perceptual qualities of the images should also align. Such texture synthesis can function as a measure for how well the defined texture model has captured features of textures. If the synthesised textures are perceptually similar to the exemplar texture, we may conclude that the statistics that were derived, capture the perceptual features well.

1.6 neural synthesis

Central to this thesis is the work by Gatys and colleagues [12]. They showed that an effective texture model can be constructed based on correlations between features from the feature space of a convolutional neural network that has been pre-trained for image classification in the ImageNet challenge. Synthesis based on their model is able to synthesise a wide variety of textures, without the need for hand-crafting image statistics.

2 STATE OF THE ART

In the last few decades, many different approaches to synthesising textures have been developed. Ranging from non-parametric patch rearrangement techniques to parametric statistical approaches. Researchers have tried to formulate models and techniques for specific classes of textures as well as for broader ranges of textures. In this chapter, we will first focus on the classical way of generating a texture model as developed by Heeger and Bergen [18] and by Portilla and Simoncelli [32]. After which we will look at the extension to this technique by Gatys et al. [12] which makes use of a set of image features learned by a convolutional neural network (CNN).

2.1 NON-PARAMETRIC APPROACHES

Exemplar-based texture synthesis is carried out by algorithmically creating a new digital image based on the perceptual nature of some example image. Exemplar-based synthesis can either be parametric or non-parametric. Some non-parametric methods that make use of, for example, tiling, patch re-arrangement, or Markov-fields have been developed [9, 14, 28]. These methods build up new textures by sampling or shuffling local areas of the original texture. While some of these methods can achieve successful synthesis results, they do not contain any theory of the statistical nature or the perceptual features of textures.

2.2 PARAMETRIC APPROACHES

A class of synthesis algorithms that does incorporate a statistical aspect are the parametric methods. These methods start by defining a texture model that tries to capture the characteristics of the texture in some sort of summary statistic that can be used to steer synthesis. Essentially, if we are able to extract the meaningful statistical information that defines human perception of texture from an image, we are then able to find other images with matching statistics that are thus perceptually equivalent to the original image [24, 25, 32].

2.2.1 *Gabor Filters*

Human perception is what determines the realised similarity of a synthesised image to its exemplar, so researchers have looked at trying to imitate the mechanism by which the visual cortex analyses visual texture. Gabor filters have been found to be a good model for two-dimensional receptive fields of simple cells in the striate cortex [21, 22]. A two-dimensional Gabor filter can be realised as a sinusoidal plane wave of some frequency and orientation within a two dimensional Gaussian envelope. When a set of Gabor filters are applied to a variety of textures found to be pre-attentively discriminable, they produce first-order differences for differently textured regions. This ability suggests that certain texton types — local features described by Julesz [24] — can be detected by application of Gabor filters [37].

Gabor filters that mimic the neurons of the visual cortex can be applied on images to produce meaningful statistics for parametrically modelling texture perception. Based on this notion, various techniques of modelling early visual processing in mammals by implementing multi-scale analysis using Gabor filters have been developed [2, 30, 37].

2.2.2 *Multi-Scale Analysis*

Heeger and Bergen [18] extended the approach to multi-scale analysis. Their work was also motivated by a model of human texture perception. They noted that textures are not easily discriminated when they produce a similar distribution of responses in a bank of (orientation and spatial-frequency selective) linear filters. Their method synthesises textures by matching the outputs of these filters. This model thus depends on the characterising information of a texture being captured in the first-order statistics of an appropriately chosen set of linear filters. However, they acknowledged limitations in their approach. The model captures some, but not all, perceptually relevant structures of natural textures. It is particularly reliant on the input image being a homogeneous texture. Inhomogeneous inputs, such as those with intensity gradients or perspective distortions, result in synthetic textures with a blotchy appearance. The approach also struggles with quasi-periodic textures and random mosaic textures, where the results, although interesting, do not closely resemble the inputs. These limitations are attributed to the model's inability to capture long-range statistical correlations, particularly in textures with varying orientations or extended fine structures.

2.2.3 *Portilla and Simoncelli*

The texture model developed by Portilla and Simoncelli [32] can be viewed as an extension to the work of Heeger and Bergen [18]. Their model also decomposes an image into oriented sub-bands of spatial frequencies using a linear filter bank. What sets this model apart from previous work, are the carefully selected joint statistical constraints.

These constraint statistics were selected by hand, based on manual visual evaluation.

They commence by selecting a set of basic parameters for a texture model and synthesised a large set of textures. After initial synthesis they identify a class of textures containing features that produced the poorest results and chose a new constraint that would capture the missing features in the model. After extending the constraint functions they re-synthesise the failed class of textures to verify the new constraint worked as intended. Furthermore, they verify if the original constraints were still necessary.

Finally, they arrive at a set of 710 statistical constraints to parameterise their texture model. The set is divided into four classes, each of which contributes to certain perceptual features. First, there are a handful of marginal statistics, followed by raw coefficient statistics that contribute to the regularity and the salient spatial frequencies, a set of coefficient magnitude statistics that represent structures such as edges and corners, and finally cross-scale phase statistics that distinguish edges from lines and can represent gradients due to shading.

It is the resourceful and specific application of auto- and cross-correlation on the filter responses that generated the rich descriptive set of constraint statistics that fully describes their texture model. The synthesis results are generally not easily discriminated in pre-attentive examination and for more than a decade represented the state of the art in texture synthesis [33].

2.3 gatys' method

In 2015, Gatys et al. [12] proposed a novel texture model based on the feature maps of VGG-19 [35], a pre-trained convolutional neural network that is optimised for object recognition. Synthesis results of the model exhibit remarkable perceptual quality, showcasing the impressive generative capabilities of neural networks trained on classification tasks. Additionally, these results may provide valuable insights into the learned representations within a neural network. Before we get into the actual synthesis method, we will first cover some deep learning concepts that are crucial to its functioning.

2.3.1 *Deep Learning*

Deep learning is a subclass of machine learning algorithms that employs a network containing multiple layers to progressively extract higher-level features from some input; each layer learns to transform its input into a slightly more abstract representation. The layers are made up of nodes/neurons, each of which has a weight and a bias that determine how it passes through information. An important feature of deep learning is that these weights and biases are learned autonomously.

2.3.1.1 *Fundamentals*

Neural networks consist of layers, each represented by a matrix of weights. The transformation of information within the network is achieved by matrix multiplication of these weights with the layer's input. Mathematically, if we denote a layer as L with weights W and input x, the output y can be represented as

$$y = Wx + b\,, \tag{1}$$

where b is the bias vector. The output of one layer becomes the input to the next, making the network a high-dimensional differentiable function mapping input to output.

During the "learning" stage, the network's weights are optimised. This is done by defining a loss function $\mathcal{L}$ which quantifies the network's performance based on its current weights. To optimise these weights, backpropagation is employed. It involves the recursive application of the chain rule to compute the gradient of the loss function with respect to each weight. This can be represented as

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial y} \cdot \frac{\partial y}{\partial W}\,. \tag{2}$$

By adjusting the weights in the direction opposite to the gradient, the network iteratively improves its performance.

2.3.1.2 *Convolutional Neural Networks*

Convolutional Neural Networks (CNNs) are designed with the assumption that the input is an image. They contain convolutional layers acting as linear filters, with the features they detect being determined during the learning process. The non-linearity of the network is introduced through activation functions. Mathematically, a convolution operation in a CNN can be represented as

$$y = f(W * x + b)\,, \tag{3}$$

where $*$ denotes the convolution operation, W is the weight matrix (filter), x is the input, b is the bias, and f is the non-linear activation function.

For instance, in image recognition, the initial input could be a pixel matrix. Subsequent layers progressively encode higher-level features, starting from edges in the first layer to complex features like facial attributes in deeper layers. This encoding process, while often conceptually simplified, is in reality optimised through training to minimise the loss function, without explicit design of the feature space.

ILSVRC The ImageNet Large Scale Visual Recognition Challenge is a yearly competition where research teams compete to achieve the highest accuracy on several visual recognition tasks using the ImageNet dataset. Around 2011 a good classification top-5 error rate was 25%. In 2012, a deep CNN called AlexNet [27] achieved 16% and marked the start of an industry-wide artificial intelligence upsurge. In the years following this breakthrough, error rates fell to only a few percent.

2.3.1.3 *Motivation behind CNN*

In an experiment where they studied the activation of neurons in the brain of a cat, Hubel & Wiesel [19] discovered that certain neurons in the cats brain would activate based on very specific stimuli. They found that straight lines would active specific neurons based on their rotation. This implies that there are basic structures in the visual cortex that respond to certain elementary shapes.

Later research has shown that the mammalian visual system contains many different parts that feed into each other in a hierarchical manner [11, 34]. In doing so, many complex features can be built up from lower elements. In this model "simple" neurons would feed into higher-level neurons, allowing the simple shapes to combine into more complex structures. This idea gave rise to the hierarchical view of the ventral stream. Simple features are combined at higher levels to eventually create the rich visual perception we experience.

The convolutional layers in deep CNNs are set up as to be able to mimic the striate cortex. Specifically, akin to how Gabor filters were applied in earlier models, convolutional layers are able to act like receptive filters analogous to the neurons in the visual system. The power of CNNs comes from the fact that the features in convolutional layers are learned from large amounts of real-world data. Hence it can be assumed that the feature space takes on some optimal form for extracting features from the input image [39].

2.3.2 *Synthesis*

The texture synthesis method developed by Gatys et al. [12] can be seen as a modified version of the work by Portilla and Simoncelli [32]. Where Portilla and Simoncelli used a steerable pyramid to decompose the input image into sub-bands, effectively filtering using a linear filter bank, Gatys et al. use the activations of the intermediate feature maps of a pre-trained convolutional neural network, and instead of the 710 hand-chosen statistics of Portilla and Simoncelli they opted for the correlations across all activations within the feature maps.

To analyse texture, an exemplar is forwarded through the pre-trained network, but instead of looking at the classification score, the intermediate activations within the layers are collected. The correlations

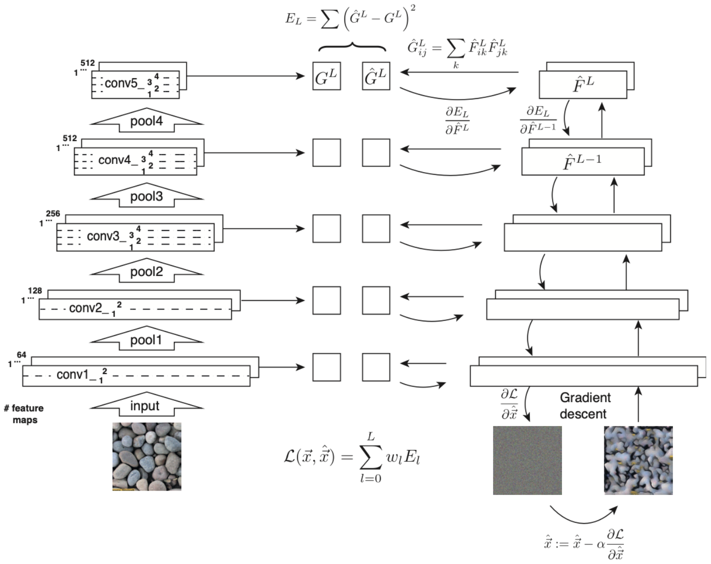

$E_L = \sum \left(\hat{G}^L - G^L\right)^2$
$\hat{G}^L_{ij} = \sum_k \hat{F}^L_{ik}\hat{F}^L_{jk}$
G^L
$\hat{G}^L$
$\hat{F}^L$
$\frac{\partial E_L}{\partial \hat{F}^L}$
$\frac{\partial E_L}{\partial \hat{F}^{L-1}}$
$\hat{F}^{L-1}$
512
1 ...
conv5_ 1 2 3 4
pool4
512
1 ...
conv4_ 1 2 3 4
pool3
256
1 ...
conv3_ 1 2 3 4
pool2
128
1 ...
conv2_ 1 2
pool1
64
1 ...
conv1_ 1 2
input
# feature maps
$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l$
$\frac{\partial \mathcal{L}}{\partial \hat{\vec{x}}}$
Gradient descent
$\hat{\vec{x}} := \hat{\vec{x}} - \alpha \frac{\partial \mathcal{L}}{\partial \hat{\vec{x}}}$

Figure 5: Illustration from the paper by Gatys et al. Texture analysis on the left. Texture synthesis on the right.

across the activations within a set of chosen feature maps are computed as a set of Gram matrices G^l (see left of Figure 5).

The same operation is performed for a seed image, for which Gatys et al. used Gaussian white noise. A loss function is defined based on the distance between the Gram matrices of the exemplar and the seed. The seed image is then adjusted using gradient descent on the total loss, as to force the feature correlations to line up with those of the exemplar (see right of 5). This process is applied until the seed image converges to a texture that is perceptually similar to the exemplar.

2.3.2.1 *VGG*

Gatys et al. selected the VGG-19 network for their application. VGG models are a type of CNN architecture proposed by Simonyan and Zisserman of the Visual Geometry Group at Oxford university [35]. They experimented with networks of different depths, of which VGG-19 is the deepest at 19 layers. Their architecture follows a basic and linear structure of convolutional layers with receptive windows of 3x3 each followed by a ReLU non-linearity [10]. They choose 3x3 filters as these are the smallest possible filter size that can still capture notions of left/right and up/down. Some convolutional layers are also followed by a max-pooling layer with 2x2 window and stride of 2. At the end of their network are three fully connected layers and a softmax layer for the classification. These last four layers are not of interest for the texture synthesis task, as only the feature maps are used. VGG achieved second place in the 2014 ILSVRC with a top-5 error of 7.3%.

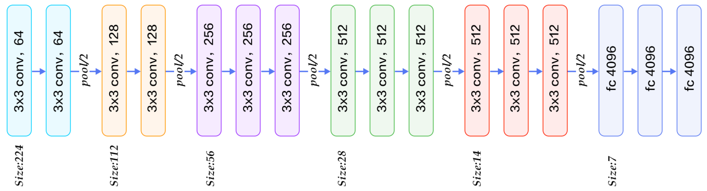


Figure 6: Example of the straightforward VGG architecture. Shown here is the VGG-16 variant.

2.3.2.2 *Gram Matrices*

When an image is forwarded through a convolutional neural network, the activations for each layer in that network can be understood as a set of filtered images, which are called *feature maps*. These maps contain the intensity of certain learned features across the image.

Central to the texture model by Gatys et al. is the use of Gram matrices. Since textures are per definition stationary, the model needs to be agnostic to the spatial information in the feature maps. A spatially invariant summary statistic is given by a Gram matrix, where the elements are the feature correlations within a layer.

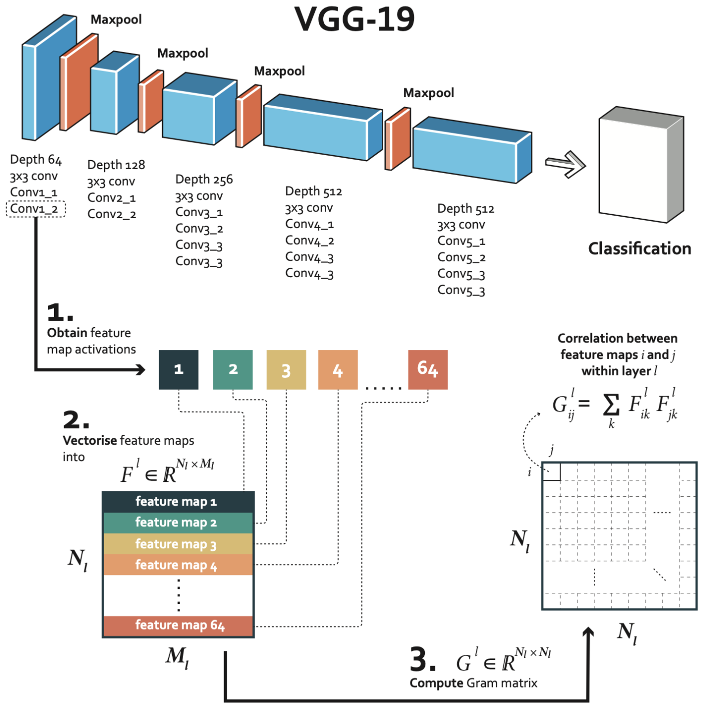


Figure 7: Computing a Gram matrix based on the feature maps of a convolutional layer.

Gram matrices are computed per layer of the network. The process start with forwarding some vectorised image $\vec{x}$ through the network and obtaining all feature maps of a layer l (see step 1 in Figure 7).

The number of feature maps N_l within a layer l is determined by the depth of that layer.

All feature maps within layer l are of equal size M_l when vectorised and can thus be stored in a matrix $F \in \mathbb{R}^{N_l \times M_l}$ (see step 2 in Figure 7).

Finally, a Gram matrix $G^l \in \mathbb{R}^{N_l \times N_l}$ can be computed (see step 3 in Figure 7), where G^l_{ij} is the dot product between feature map i and j in layer l is

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \,. \tag{4}$$

2.3.2.3 *Loss and Optimisation*

To synthesise a texture using the method as defined in [12], a white noise image is updated to make its Gram matrix representation match that of an exemplar. This process is achieved using gradient descent. The loss function is defined as the mean-squared distance between the Gram matrices obtained from the exemplar and the Gram matrices obtained from the image that is synthesised.

Let $\vec{x}$ and $\hat{\vec{x}}$ be the original image and the image that is generated, and G^l and $\hat{G}^l$ their respective Gram matrix representations in layer l. The contribution of layer l to the total loss is then

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - \hat{G}^l_{ij})^2 \,, \tag{5}$$

and the total loss is

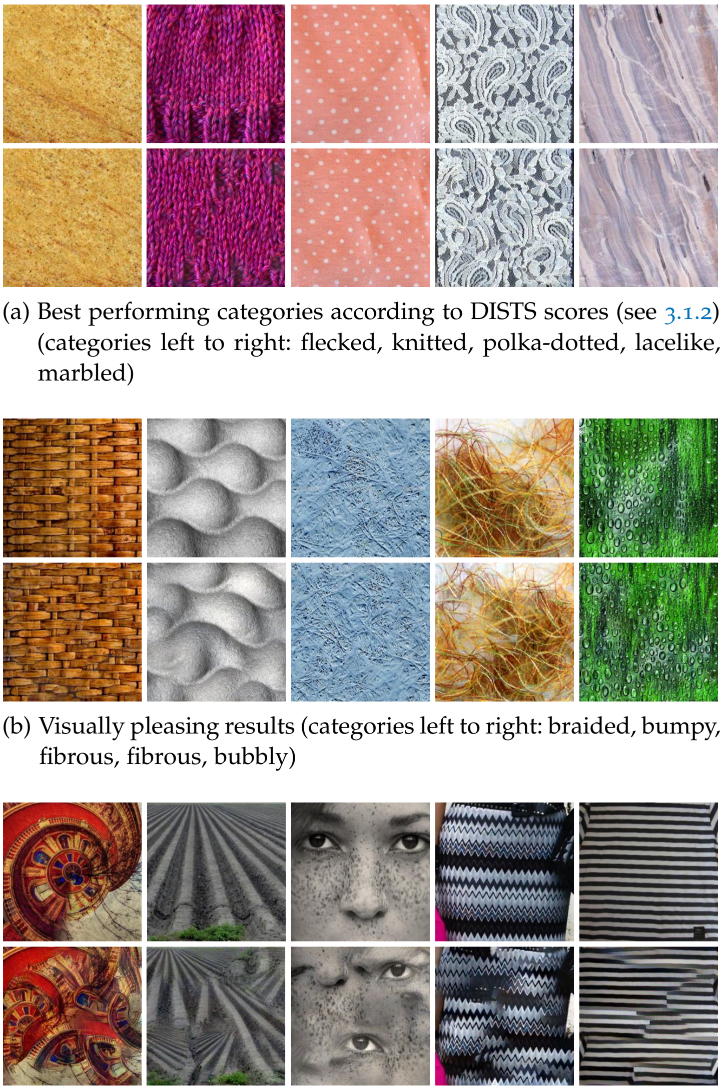
$$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l \,, \tag{6}$$

where w_l are the weighting factors of the contribution of each layer to the total loss.

2.3.2.4 *Synthesis*

The following results were obtained from a custom Python implementation of the method by Gatys et al. To facilitate ease of implementation, the PyTorch deep learning library [31] was used. All images were synthesised with a GTX 4090, which allowed for faster running times using GPU acceleration. Feature maps from all convolutional and all pooling layers in VGG-19 were used to parameterise the model. For implementation details see the GitHub repo.

We made use of the Describable Texture Dataset (DTD) [6]. This allowed for the method to be tested on a large array of varying textures. The dataset consists of 47 categories, each of which contains 120 different images. We choose 10 images at random from every category to create a subset of the original DTD. This was necessary due to the slow nature of the synthesis method. Unfortunately some images in DTD contain only small areas of their texture category and are more like regular structured images, which were not well synthesised.



(a) Best performing categories according to DISTS scores (see 3.1.2) (categories left to right: flecked, knitted, polka-dotted, lacelike, marbled)

(b) Visually pleasing results (categories left to right: braided, bumpy, fibrous, fibrous, bubbly)

(c) Synthesis often fails for structures and straight lines (categories left to right: spiralled, grooved, freckled, zigzagged, banded)

Figure 8: Synthesis results. Exemplar on top, synthesised texture on bottom.

Figure 8 shows some examples of texture categories from DTD that are generally synthesised well, while figure 8 shows examples of categories that present the model with problems. Among the worse performing textures are those containing straight lines and structure, which are not preserved well by the model. The 'freckled' category is an example of a set of images in DTD that are difficult to synthesise as only images of faces with small freckled areas are included.

Images that produce satisfying results are generally entirely composed of small-scale discernible elements repeated across the image in stochastic fashion.

3 RESEARCH

The work of Gatys et al. [12] on texture synthesis using deep convolutional neural networks has marked a significant milestone. Building upon their findings, this research explores the further potential and applications of neural synthesis in the realm of texture upscaling and enhancement.

3.1 METHODOLOGY

Our research consists of four separate objectives, each of which will be investigated independently. Nonetheless, throughout our research we use the same image quality metric as introduced in 3.1.2.

3.1.1 *Research Objectives*

This research is driven by a series of objectives and questions designed to probe deeper into the capabilities and limitations of neural synthesis in texture upscaling. The key objectives are:

1. To develop and propose a stopping criterion for the synthesis process based on loss plots.
2. To experiment with different seed images and assess their impact on the synthesis process.
3. To explore the role of self-similarity at different scales in texture upscaling.
4. To evaluate the performance of a different pre-trained network, such as ResNet [17], in texture synthesis quality.

3.1.2 *Image Quality Assessment*

In order to establish uniform quality judgements for a large set of synthesised textures, our research employs a full-reference Image Quality Assessment (IQA) method. IQA is essential for quantifying human perception of image quality, which is a critical aspect of our study on texture synthesis. For this purpose, we have selected the Deep Image Structure and Texture Similarity (DISTS) metric [8].

The DISTS method has been found to more closely match human perception of similarity than other IQA methods, such as the Structural Similarity Index Measure (SSIM) [8]. This alignment with human visual perception makes DISTS particularly suitable for our study, where the quality of texture synthesis is a subjective measure. By employing DISTS, we can make faster and more accurate inferences regarding the average quality of synthesised textures, enabling us to feasibly compare different synthesis techniques.

To effectively interpret DISTS scores in the context of our research, we conducted a preliminary bench-marking exercise. This involved comparing all images in the Describable Textures Dataset (DTD) with Gaussian noise. Given that our synthesis process begins with Gaussian noise, using it as a benchmark provides a relevant and meaningful assessment for the DISTS metric. Through this comparison across the entire DTD, we obtained a mean DISTS score of 0.338 with a standard deviation of 0.043. This benchmark will serve as a reference point in our subsequent analysis, allowing us to gauge the effectiveness of our texture synthesis approach in comparison to a known baseline.

3.2 DEFINING A STOPPING CRITERION

Establishing a precise stopping criterion is crucial within any context which involves the optimisation of some loss function, as it directly influences the computational efficiency and the final quality of the output. A basic approach might rely on a predetermined iteration count. However, the optimisation process might reach some desired state well before this iteration limit. To overcome this limitation, we propose a stopping criterion based on loss quantity. Our approach hinges on the utilisation of loss plots generated during the synthesis process, as proposed by Gatys et al. [12] By analysing these plots, we aim to identify a stopping point that balances the fidelity of the synthesised texture against the computational expenditure.

3.2.1 *Methodology*

The methodology revolves around analysis of the loss plots, which are graphical representations of the loss function values over the course of the synthesis iterations. These plots provide insights into the convergence behaviour of the synthesis process. We will examine the characteristics of these plots, such as the rate of change in loss, the presence of plateaus, and any patterns that emerge over iterations.

For each texture synthesis process, we will record the loss at every optimisation step. This data will be used to construct loss plots that capture the progression of the synthesis over time.

Our analytical approach involves identifying key features in the loss plots that signal the nearing of an optimal stopping point. This could include identifying points of diminishing returns, where further iterations result in negligible improvements in loss, or detecting points of stability, where the loss value remains consistent over a significant number of iterations.

Based on the insights gained from the loss plot analysis, we will propose a single stopping criterion in the form of a loss function measure. A longer synthesis process will in general lead to better quality textures, and a stopping criterion will by it's nature eliminate some of that potential. However, by choosing the criterion so synthesised textures are still able to achieve perceptual similarity to their exemplar (verified by visual inspection), we ensure a balance between synthesis quality and computational cost.

To validate the effectiveness of the proposed stopping criterion, it will be applied to a subset of the Describable Textures Dataset. The synthesis results at these algorithmically determined stopping points will be compared against results obtained using a maximum iteration count to assess improvements in efficiency and quality.

3.2.2 *Analysis of Loss Plots*

In searching for a stopping criterion for texture synthesis, an analysis of loss plots from various texture categories was undertaken. This analysis was instrumental in identifying a common pattern across different textures: the gradient of the loss plot, representing the rate of change of the synthesis loss, tends to flatten as the synthesis progresses. This flattening of the gradient is indicative of diminishing returns, where subsequent iterations contribute less significantly to the improvement of the synthesised texture.

One critical observation from our analysis was the variability in the point at which the loss gradient begins to flatten. The top-left lossplot in Figure 9 stabilises just above 1, while the top-right lossplot stabilises at a far lower value. This variance indicates that the synthesis process behaves differently across various textures, possibly due to the inherent complexities and characteristics unique to each texture. Furthermore, for lossplots like the bottom examples in Figure 9, earlier termination would have been beneficial, as the loss function shoots up near the end.

Initially, the idea of implementing a fixed threshold for the loss value as a stopping criterion was considered. However, the observed variability in the loss plot behaviours suggested that a fixed threshold would lack the necessary flexibility to be universally applicable across different types of textures. A predetermined loss value might be too high for some textures, leading to premature termination, or too low for others, resulting in unnecessary computational expenditure.

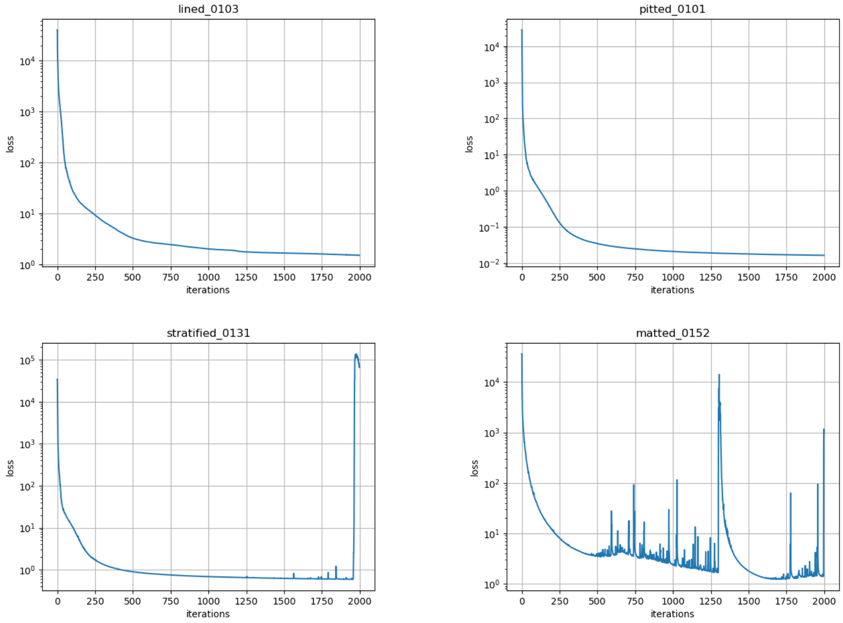


Figure 9: Comparative analysis of lossplots

3.2.3 *Proposed Gradient Threshold*

In light of these observations, we pivot towards a more dynamic approach: setting a gradient threshold as the stopping criterion. This method focuses on the rate at which the loss decreases, rather than the absolute loss value. The gradient threshold is defined as the point at which the slope of the loss plot reaches a level of flatness, suggesting that further iterations are unlikely to yield significant improvements in the quality of the synthesised texture.

We define the gradient of the loss function, G, as the average rate of change of the loss over a specified window of iterations. This is calculated as

$$G = \frac{L_{t-w+1} - L_t}{w}, \tag{7}$$

where:

- L_t is the loss at the current iteration t,
- w is the window size, and
- L_{t-w+1} is the loss at the iteration $t - w + 1$.

The stopping criterion for the synthesis process is then determined by the magnitude of the gradient G. The process is terminated when the absolute value of G falls below a predefined threshold, θ, indicating that the rate of improvement in the loss has decreased to a point of diminishing returns. This is expressed as

$$|G| < \theta\,, \tag{8}$$

where θ is the stopping threshold. By employing this criterion, the synthesis process is dynamically adjusted to each texture, terminating when further iterations are unlikely to yield significant improvements. This approach ensures both computational efficiency and the maintenance of synthesis quality.

3.2.4 *Threshold Analysis*

To determine an appropriate gradient threshold, we experimented with thresholds in different orders of magnitude and manually inspected the visual quality versus computational cost. By examining the results in Figure 10 we can determine that a gradient threshold greatly reduces the computational cost while maintaining quality. An interesting observation is that by implementing a threshold of 0.1 we obtained a higher DISTS score with fewer steps needed than without a threshold. Upon closer inspection of results generated without a threshold, we noticed that a reasonable amount of these lossplots exhibited patterns like those in the bottom row of Figure 9. In such cases it seems like the optimiser is able to find a local minimum, but at some points drastically overshoots and terminates in a worse state. By applying a gradient threshold to the loss function, we are able to terminate whenever the loss value stabilises and prevent such overshoots, resulting in an on average higher DISTS score.

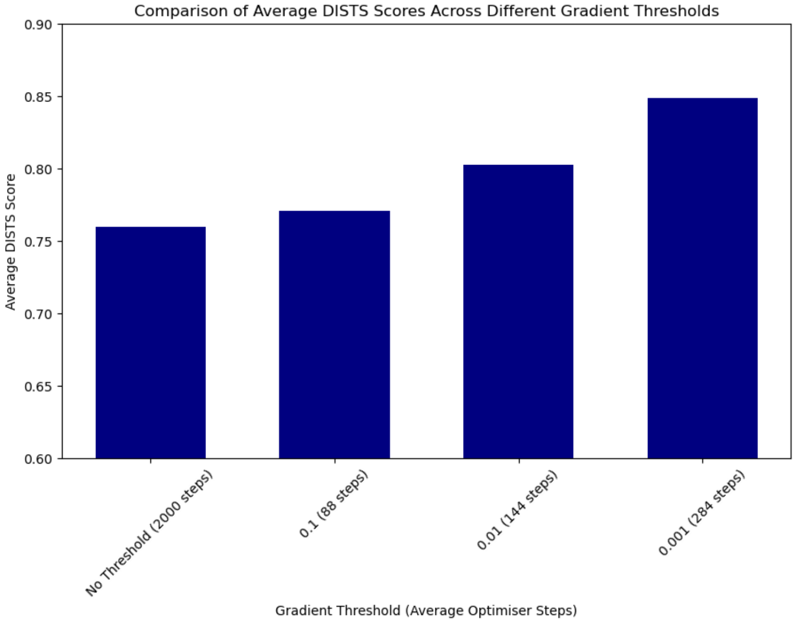


Figure 10: Analysis of loss gradient thresholds

3.3 SEED IMAGE VARIETIES

In addition to using high-frequency Gaussian noise as a seed image for texture synthesis, we aim to explore the impact of alternative seed images, particularly focusing on blurred exemplars, low-frequency noise and distinct geometric patterns such as lines or chequerboards. This investigation stems from the hypothesis that different initial conditions in the synthesis process can significantly influence the final texture quality, especially in terms of preserving or enhancing certain structural elements within the images.

3.3.1 *Methodology*

To systematically examine the effects of these alternative seed images, the same DTD subset is used, synthesising textures with each type of seed image and subsequently recording the DISTS scores. By comparing the results obtained with low-frequency noise and geometric patterns against those achieved with blurred exemplars and white noise, we aim to draw comprehensive insights into the influence of seed image characteristics on texture synthesis.

3.3.2 *Gaussian Noise*

In the work by Gatys et al. [12], Gaussian noise serves as the starting point for the texture synthesis process. Gaussian noise, often referred to as white noise, is characterised by its probability distribution, where each pixel in the image is assigned a value according to the Gaussian, or normal distribution. This distribution is defined by two key parameters: the mean (usually zero) and the standard deviation (which determines the spread or variance). In our context, Gaussian noise is generated with a mean of zero and a standard deviation of one, resulting in a normal distribution centred around zero.

An essential aspect of Gaussian noise is its representation in the frequency domain. Gaussian white noise is evenly distributed across all frequencies. This even distribution implies that no specific frequency is favoured, ensuring that the synthesis process is not biased towards any particular pattern or structure at the outset.

3.3.3 *Blurred Exemplar*

In the frequency domain, Gaussian blur acts as a low-pass filter, attenuating high-frequency components more than low-frequency components. This ability to retain the macro-structure of the original texture by preserving low-frequency components provides the rationale for using blurred exemplars as seed images. This method is hypothesised to assist the synthesis model in capturing the broader patterns and general outlines of the texture. Such an approach could be particularly beneficial for textures where the overall structure is more significant than fine details.

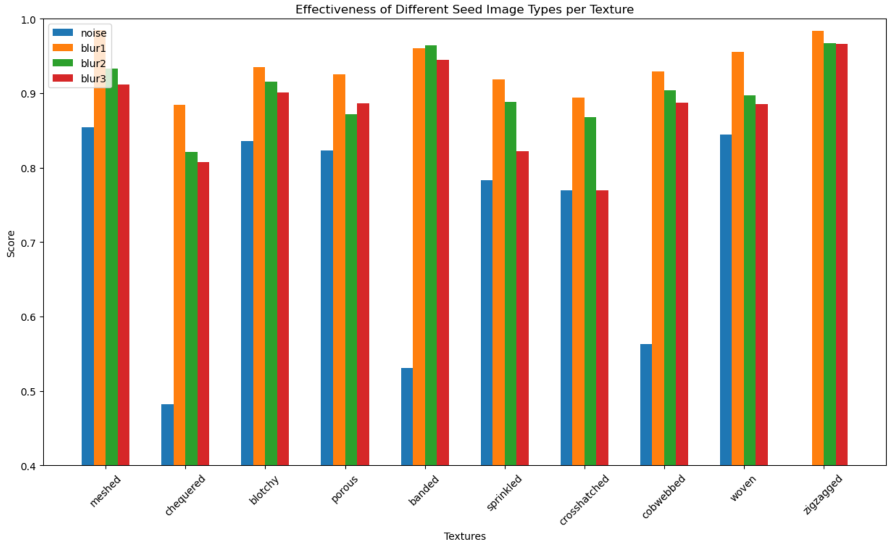


Figure 11: Different blur intensities as seed images

Our analysis in Figure 11 clearly indicates that the use of blurred exemplars results in better DISTS scores, affirming the expectation that incorporating low-frequency components at the onset of the synthesis process enhances the overall quality. This is especially pronounced in textures where synthesis with white noise as a seed image traditionally struggles, such as "chequered", "banded", "cobwebbed", and "zigzagged".

We note that for these texture classes, which are predominantly characterised by their low-frequency patterns, the transition to synthesis from a blurred exemplar exhibits significant improvements. This observation underscores a key aspect of Gatys et al.'s [12] method — its proficiency in synthesising high-frequency components. Thus effectiveness is enhanced when the synthesis process begins with a seed image that already includes the underlying low-frequency patterns. This synergy between the low-frequency starting point and the method's high-frequency synthesis capabilities leads to a more accurate and visually pleasing recreation of the original textures.

3.3.4 *Low-Frequency Noise*

Inspired by the blurred exemplar seed images, we decide to investigate low-frequency noise as low-frequency patterns could provide a basic structural framework, potentially aiding the model in capturing and replicating these broader patterns more effectively. This approach contrasts with the high-frequency randomness of white noise, which might overlook or inadequately reproduce such large-scale features.

We produce low-frequency noise by applying a Gaussian blur over a white noise image, attenuating the high-frequency components.

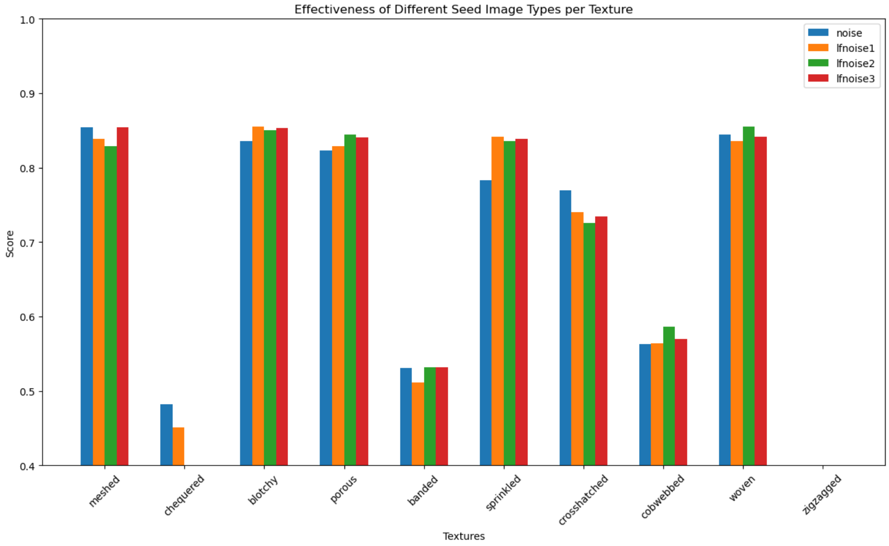


Figure 12: Different low-frequency noise intensities as seed images

Our investigation into the use of low-frequency noise as a seed image for texture synthesis, specifically white noise subjected to Gaussian blur, revealed no significant structural improvements over the use of regular white noise. This outcome (Figure 12) suggests that the mere presence of low-frequency information in the seed image is insufficient to enhance the synthesis process. Crucially, it appears that it is not the low-frequency components per se that are beneficial, but rather the specific alignment of these components with the structures inherent in the exemplar. The random nature of the low-frequency noise, despite its blurred characteristics, did not contribute positively, proving no more effective than Gaussian white noise as a seed image for synthesis.

3.3.5 *Geometric Patterns*

Similarly, geometric patterns like lines or chequerboards are selected to assess their utility as seed images. The regularity and distinctness of these patterns present a unique opportunity for the synthesis process. For instance, starting with a chequerboard pattern might influence the model's ability to maintain or transform regular geometric structures within the synthesised textures. We expect such seed images to mainly benefit the very regular texture types.

We select three types of geometric patterns: horizontal lines, vertical lines, and a chequerboard. For each type we generate seed images using line and square widths of two, four, and eight.

Inspection of the results in Figure 13 reveals that none of the geometric patterns have any notable advantage, except for the special

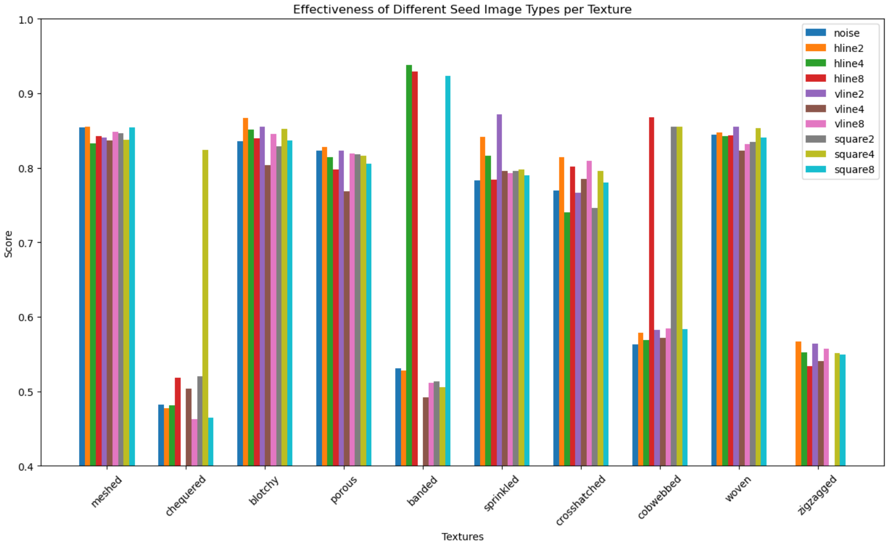


Figure 13: Lines and squares as seed images

cases where the regular high-frequency textures of the "chequered", "banded", and "cobwebbed" classes happened to align with the geometric seed.

3.4 SCALE INVARIANCE OF TEXTURES

Scale invariance, or the property by which a texture exhibits similar patterns at different scales, is a fundamental characteristic in many natural textures. This aspect of our investigation into texture synthesis revolves around the concept of scale invariance, particularly its role and potential benefits in the context of texture upscaling at varying scales. The primary objective of this analysis is to establish a metric of scale invariance for each image, which will then be used to investigate the potential correlation between levels of self-similarity and the quality of texture synthesis.

3.4.1 *Methodology*

We employ a wavelet transform to perform a multi-scale analysis on each texture. This approach enables us to quantify the scale-invariance characteristic of textures, a metric that we hypothesise would correlate with the DISTS scores, indicative of perceptual similarity. Wavelet decomposition allows for the analysis of an image at various scales, capturing both frequency and location information. The methodology involved the following steps:

1. **Wavelet Decomposition:** We use the PyWavelets library to perform wavelet decomposition. The decomposition is executed as follows:

- Determine the maximum level of decomposition, based on the minimum dimension of the image and the decomposition length of the chosen wavelet.
- Apply the wavelet decomposition to obtain a set of coefficients at various scales. We select the Haar wavelet for its simplicity and efficiency.

2. **Scale-Invariance Metric Calculation:** The scale-invariance metric is computed by analysing the wavelet coefficients at different scales. Specifically, the mean of the absolute values of the coefficients, excluding the approximation coefficients, is calculated. This mean value serves as an aggregate score representing the scale-invariance of the texture.

3.4.2 *Statistical Analysis*

We seek to understand the relationship between perceptual similarity of the synthesised image, as measured by DISTS scores, and a scale-invariance metric. To this end, we employed Pearson's correlation coefficient, a statistical measure that quantifies the linear relationship between two variables. Our analysis yields a Pearson correlation coefficient of 0.1855 and a p-value of 8.42×10^{-5}.

The Pearson coefficient of 0.1855 indicates a slight, yet positive linear relationship between the DISTS scores and the scale-invariance metrics. This suggests that as the scale-invariance metric increases, there is a mild tendency for the DISTS scores to increase as well. However, the relatively low value of the coefficient implies that this relationship is weak. It indicates that other factors not accounted for in this analysis might also play a significant role in determining the DISTS scores.

Importantly, the p-value associated with this correlation coefficient is 8.42×10^{-5}, which is well below the commonly used significance level of 0.05. This low p-value indicates that the observed correlation is statistically significant and unlikely to have occurred by random chance. Thus, while the strength of the relationship is weak, we can be reasonably confident that the correlation between these two variables is not coincidental.

In conclusion, our findings suggest a statistically significant, albeit weak, positive linear relationship between perceptual similarity and scale-invariance in texture synthesis.

3.5 TEXTURE SYNTHESIS USING RESNET FEATURES

The work by Gatys et al. [12] utilised the feature space of VGG-19 in their texture synthesis model. In our research, we extend this inquiry to explore whether such synthesis capabilities can be generalised to other network architectures. Specifically, we focus on the residual network (ResNet), a framework proposed by researchers at Microsoft [17], designed to train deeper neural networks more effectively. Traditional deeper networks faced challenges in converging during stochastic gradient descent with backpropagation, often attributed to vanishing gradients [1]. This issue was substantially mitigated through strategies such as normalised initialisation [13] and the integration of intermediate normalisation layers [20]. However, even with these advancements, deeper networks tended to exhibit saturation and rapid degradation in accuracy as more layers were added, leading to higher training errors [16, 17].

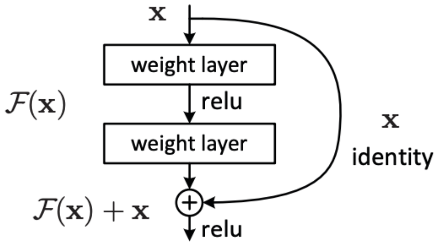

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 14: A residual block

Residual learning, proposed to address the accuracy degradation in deeper networks, introduces an innovative approach. Rather than having a few stacked layers learn an underlying mapping directly, these layers are tasked with learning a residual mapping. The practical implementation of this concept employs "shortcut connections" that skip one or more layers, performing identity mapping when added to the output of stacked layers (as illustrated in Figure 14). These shortcuts contribute neither additional parameters nor complexity.

3.5.1 *Architecture*

ResNet architecture comes in various configurations, with different layer depths. Its baseline network, used for creating a residual network by adding shortcut connections, shares similarities with the structure of VGG networks, predominantly using 3x3 convolutional filters. Downsampling within the network is directly achieved by convolutional layers with a stride of 2. Pre-trained ResNet models are available in PyTorch with varying depths, including 18, 34, 50, 101, and 152 layers. For our texture synthesis evaluation, we selected the 34-layer variant, known as ResNet34. This architecture, while mirroring the VGG architecture in several aspects, distinguishes itself with the inclusion of skip connections.

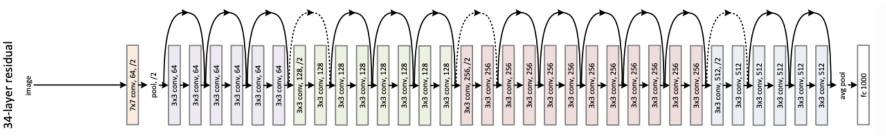


Figure 15: ResNet34

3.5.2 *Feature Space*

In the context of ResNet, the desired mapping for a part of the network can be denoted as $\mathcal{H}(x)$. Hence, the layers within a residual block are designed to learn a mapping $\mathcal{F}(x) = \mathcal{H}(x) - x$. Consequently, we suggest extracting the feature space from the output immediately after the identity is added, which equates to the original mapping $\mathcal{H}(x) = \mathcal{F}(x) + x$. Our texture model thus utilises the feature space derived from every residual block in ResNet34, specifically from the point after the final non-linearity in each block.

3.5.3 *PyTorch Implementation*

To access the feature maps from ResNet34, we implement a custom class in PyTorch that inserts a 'forward hook' after each residual block. This forward hook, activated during the network's forward pass, captures both the input and output of a given layer, thereby enabling the extraction and storage of raw feature maps as an image progresses through the network. The implementation methodology aligns closely with the one used for VGG, involving the computation of Gram matrices from the feature maps. The total loss is calculated by summing the mean-squared distances between corresponding Gram matrices from matching layers. The optimisation process involves gradient descent using L-BFGS [29], starting from an image of white noise. We set an iteration limit of 2000 and implement a stopping criterion when the total loss falls below 10^{-3}.

3.5.4 *Results*

To rigorously evaluate the capability of ResNet34's feature space in parameterising texture models, our study initiates by synthesising textures from both the VGG and ResNet34 networks. This process involves selecting a diverse subset from the DTD dataset, encompassing 10 representative images from each category. The primary focus was to assess the synthesis quality of these textures, quantitatively measured using the DISTS scores, and subsequently comparing them against their original exemplars. With the well-established proficiency of VGG-19 in texture synthesis serving as our benchmark, we sought to comprehensively understand and quantify the relative performance of ResNet34.

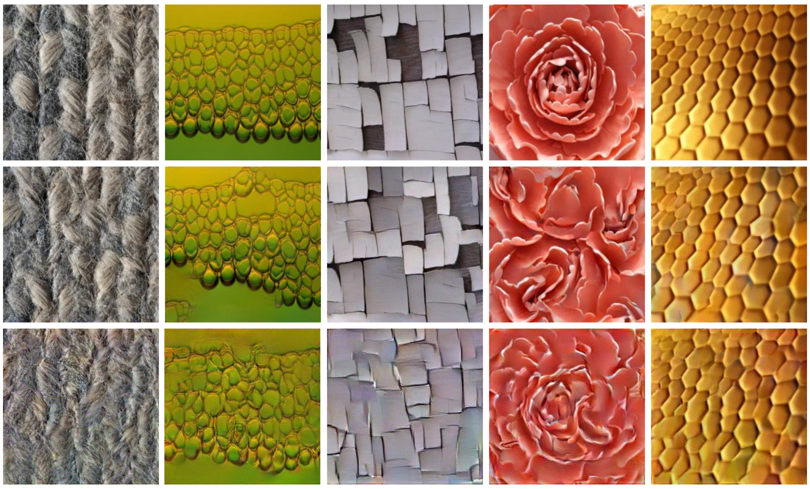


Figure 16: Comparative visual analysis of textures synthesised using VGG-19 and ResNet34 (top: exemplar, middle: VGG-19, bottom: ResNet34)

In Figure 16, we provide a visual juxtaposition of a select group of textures synthesised using both networks. This comparison underscores the observable similarities between the results, whilst also highlighting VGG-19's superior ability in preserving structural details and shadow nuances.

A more granular analysis, as shown in Figure 17, offers a category-wise comparison of the synthesis results. It reveals that, although the synthesis via VGG-19 generally leads to higher DISTS scores, the scores attained with ResNet34 notably exceed the baseline average score of Gaussian noise, recorded at 0.338. This disparity in scores, while indicative of VGG-19's potential alignment with the DISTS metric – a byproduct of its basis on VGG-16 – also underscores the proficiency of ResNet34 in generating visually coherent textures. Notably, the textures synthesised using ResNet34, while slightly inferior in quality to those from VGG-19, still demonstrate a significant correlation with human visual perception, as postulated in previous studies on the emergent properties of deep features as perceptual metrics [39]. These findings compellingly suggest that the texture synthesis methodology developed by Gatys et al. retains its effectiveness even when adapted to a different feature space, as exemplified by ResNet34.

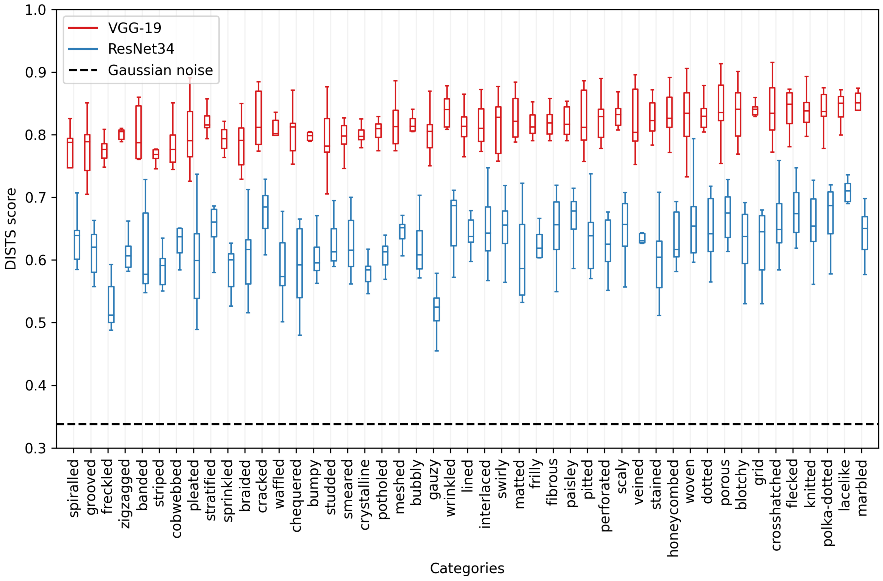


Figure 17: Category-wise comparison of DISTS scores between VGG-19 and ResNet34

4 CONCLUSION

4.1 DISCUSSION

We started by proposing a gradient threshold as a stopping criterion for the synthesis process. This approach addressed the challenge of determining the optimal stopping point, especially given the diminishing returns observed in loss value decrease. Implementing this gradient threshold significantly improved the computational efficiency of the synthesis process without compromising the quality of the synthesised textures.

Our investigation extended to evaluating the effectiveness of different seed image types, primarily chosen based on their frequency domain content. This analysis provided insights into how the choice of seed image influences the synthesis process and its outcomes. Notably, we found that seed images with low-frequency characteristics aligning closely with those of the target exemplar were more effective in guiding the synthesis process.

Furthermore, we explored the relationship between scale invariance in textures and their synthesis quality. Using wavelet transform for multi-scale analysis, we calculated a scale-invariance metric for each texture and examined its correlation with DISTS scores. This study revealed a nuanced relationship, suggesting that textures with higher scale invariance may yield better synthesis outcomes.

Finally, our research aimed to explore several key aspects of the neural synthesis method by Gatys et al. [12], focusing on its generalisability to different convolutional neural networks (CNNs). Through careful extraction of feature maps corresponding to those in VGG-19, we demonstrated that texture synthesis based on ResNet34's feature space is feasible. This was evidenced by the DISTS scores of synthesised textures, which generally fell between 0.6 and 0.7, underscoring the robustness and adaptability of Gatys' method. In summary, our research has extended the method by Gatys to the feature space of ResNet34, demonstrating its adaptability and the effectiveness of CNN weights in texture synthesis.

4.1.1 *Interpretation*

Our examination of various seed images, particularly in terms of their frequency domain content, sheds light on the impact of initial conditions on the synthesis outcome. The efficacy of low-frequency seed images, especially those mirroring the target exemplar's structure, underscores the importance of aligning the seed image's characteristics with the desired texture.

The exploration of the relationship between scale invariance and synthesis quality, facilitated by wavelet transform analysis, reveals a nuanced interplay between texture properties and perceived quality. The correlation between higher scale invariance and improved synthesis outcomes suggests that textures exhibiting consistent patterns across scales are more amenable to high-quality synthesis.

Extending the synthesis method by Gatys et al. [12] to the feature space of ResNet34 provided a robust test of its generalisability. Our results indicate that the method is not confined to the specific architecture of VGG networks but can be successfully applied to other high-performing CNNs. This adaptability, coupled with the effectiveness of ResNet34's feature space in achieving comparable DISTS scores, reaffirms the importance of CNN weights in neural texture synthesis.

4.1.2 *Limitations*

Gatys [12] method, despite its efficacy in handling a wide range of textures, showed a marked difficulty in synthesising textures with highly regular and low-frequency components. These textures often presented a unique challenge, resulting in outcomes that were either poorly synthesised or, in some instances, a complete failure of the synthesis process. This limitation was most pronounced in textures where the defining characteristics were predominantly large, uniform patterns such as artificial straight lines or zigzag patterns, which the method struggled to replicate accurately.

4.2 conclusion

Our research confirms the adaptability of Gatys et al.'s method to different CNNs, as evidenced by the successful application to ResNet34, highlighting the potential of diverse CNN architectures in texture synthesis. The implementation of a gradient threshold as a stopping criterion has demonstrated a significant improvement in computational efficiency without sacrificing the quality of the synthesised textures. Furthermore, the exploration of different seed image types, especially in relation to their frequency domain content, has provided valuable insights into the synthesis process. Finally, we have shown there is a possible positive correlation between scale-invariance within a texture and its neural synthesis quality.

4.3 future work

Our use of a basic measure for scale invariance presents an opportunity for refinement and expansion. Future studies could focus on developing a more sophisticated scale-invariance metric that captures a wider range of texture properties. A refined metric could provide a deeper understanding of the relationship between texture characteristics and perceptual quality, leading to more nuanced analyses and potentially informing the development of improved synthesis algorithms. Another direction for research is the exploration of Gatys et al.'s [12] method across a broader spectrum of neural networks. Investigating the applicability and effectiveness of this synthesis method with other CNN architectures, or even a combination of multiple networks, could reveal new dimensions of the method's capabilities. Such studies could lead to novel synthesis approaches that leverage the unique strengths of different networks, potentially resulting in more versatile and powerful texture synthesis solutions.

BIBLIOGRAPHY

[1] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult." In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[2] James R. Bergen and Edward H. Adelson. "Visual texture segmentation based on energy measures." In: *Annual Meeting Optical Society of America* (1986).

[3] James Bergen and Michael Landy. "Computational Modeling of Visual Texture Segregation." In: 1 (Aug. 1997).

[4] Charles Bordenave, Yann Gousseau, and François Roueff. "The Dead Leaves Model: A General Tessellation Modeling Occlusion." In: *Advances in Applied Probability* 38.1 (2006). Full publication date: Mar., 2006, pp. 31–46. ISSN: 00018678. URL: http://www.jstor.org/stable/20443426.

[5] T. Caelli and B. Julesz. "On perceptual analyzers underlying visual texture discrimination: Part I." In: *Biological Cybernetics* 28.3 (1978), pp. 167–175. ISSN: 1432-0770. DOI: 10.1007/BF00337138. URL: https://doi.org/10.1007/BF00337138.

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. "Describing Textures in the Wild." In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[7] George R. Cross and Anil K. Jain. "Markov Random Field Texture Models." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5.1 (1983), pp. 25–39. DOI: 10.1109/TPAMI.1983.4767341.

[8] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. "Image Quality Assessment: Unifying Structure and Texture Similarity." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.5 (2022), pp. 2567–2581. DOI: 10.1109/TPAMI.2020.3045810.

[9] A.A. Efros and T.K. Leung. "Texture synthesis by non-parametric sampling." In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1033–1038 vol.2. DOI: 10.1109/ICCV.1999.790383.

[10] Kunihiko Fukushima. "Cognitron: A self-organizing multilayered neural network." In: *Biological Cybernetics* 20.3 (1975), pp. 121–136. ISSN: 1432-0770. DOI: 10.1007/BF00342633. URL: https://doi.org/10.1007/BF00342633.

[11] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.” In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[12] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “Texture Synthesis Using Convolutional Neural Networks.” In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’15. Montreal, Canada: MIT Press, 2015, 262–270.

[13] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks.” In: *International Conference on Artificial Intelligence and Statistics*. 2010.

[14] Baining Guo, Heung yeung Shum, and Ying-Qing Xu. “Chaos Mosaic: Fast and Memory Efficient Texture Synthesis.” In: 2000.

[15] R.M. Haralick. “Statistical and structural approaches to texture.” In: *Proceedings of the IEEE* 67.5 (1979), pp. 786–804. DOI: 10.1109/PROC.1979.11328.

[16] Kaiming He and Jian Sun. “Convolutional neural networks at constrained time cost.” In: June 2015, pp. 5353–5360. DOI: 10.1109/CVPR.2015.7299173.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition.” In: June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[18] D.J. Heeger and J.R. Bergen. “Pyramid-based texture analysis/synthesis.” In: *Proceedings., International Conference on Image Processing*. Vol. 3. 1995, 648–651 vol.3. DOI: 10.1109/ICIP.1995.537718.

[19] D H Hubel and T N Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex.” en. In: *J Physiol* 160.1 (Jan. 1962), pp. 106–154.

[20] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, 448–456.

[21] J P Jones and L A Palmer. “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex.” en. In: *J Neurophysiol* 58.6 (Dec. 1987), pp. 1233–1258.

[22] J P Jones and L A Palmer. “The two-dimensional spatial structure of simple receptive fields in cat striate cortex.” en. In: *J Neurophysiol* 58.6 (Dec. 1987), pp. 1187–1211.

[23] B Julesz. "A theory of preattentive texture discrimination based on first-order statistics of textons." en. In: *Biol Cybern* 41.2 (1981), pp. 131–138.

[24] Béla Julesz. "Visual Pattern Discrimination." In: *IRE Transactions on Information Theory* 8.2 (1962), pp. 84–92. DOI: 10.1109/TIT.1962.1057698.

[25] Béla Julesz. "Experiments in the visual perception of texture." In: *Scientific American* 232 4 (1975), pp. 34–43.

[26] Béla Julesz. "Textons, the elements of texture perception, and their interactions." In: *Nature* 290 (1981), pp. 91–97.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Commun. ACM* 60.6 (2017), 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[28] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. "Real-Time Texture Synthesis by Patch-Based Sampling." In: *ACM Trans. Graph.* 20.3 (2001), 127–150. ISSN: 0730-0301. DOI: 10.1145/501786.501787. URL: https://doi.org/10.1145/501786.501787.

[29] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization." In: *Mathematical Programming* 45.1 (1989), pp. 503–528. ISSN: 1436-4646. DOI: 10.1007/BF01589116. URL: https://doi.org/10.1007/BF01589116.

[30] J Malik and P Perona. "Preattentive texture discrimination with early vision mechanisms." en. In: *J Opt Soc Am A* 7.5 (May 1990), pp. 923–932.

[31] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[32] Javier Portilla and Eero P. Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients." In: *International Journal of Computer Vision* 40.1 (2000), pp. 49–70. ISSN: 1573-1405. DOI: 10.1023/A:1026553619983. URL: https://doi.org/10.1023/A:1026553619983.

[33] Lara Raad, Axel Davy, Agnès Desolneux, and Jean-Michel Morel. "A survey of exemplar-based texture synthesis." In: *CoRR* abs/1707.07184 (2017). arXiv: 1707.07184. URL: http://arxiv.org/abs/1707.07184.

[34] Maximilian Riesenhuber and Tomaso Poggio. "Computational Models of Object Recognition in Cortex: A Review." In: (June 2001).

[35] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[36] Richard Taylor. "Chapter 11 - Fractal Expressionism—Where Art Meets Science." In: *Art and Complexity*. Ed. by John Casti and Anders Karlqvist. Amsterdam: JAI, 2003, pp. 117–144. ISBN: 978-0-444-50944-4. DOI: https://doi.org/10.1016/B978-044450944-4/50012-8. URL: https://www.sciencedirect.com/science/article/pii/B9780444509444500128.

[37] M. R. Turner. "Texture discrimination by Gabor functions." In: *Biological Cybernetics* 55.2 (1986), pp. 71–82. ISSN: 1432-0770. DOI: 10.1007/BF00341922. URL: https://doi.org/10.1007/BF00341922.

[38] Jarke J. van Wijk. "Spot Noise Texture Synthesis for Data Visualization." In: *SIGGRAPH Comput. Graph.* 25.4 (1991), 309–318. ISSN: 0097-8930. DOI: 10.1145/127719.122751. URL: https://doi.org/10.1145/127719.122751.

[39] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV].