Home Staging Using Machine Learning Techniques

By

Martí Grau Gasulla
Supervisors: *Javier Ruiz Hidalgo, Eduard Alarcon*

Department of Image
UNIVERSITAT POLITECNICA DE CATALUNYA

A thesis submitted for the Degree of
TELECOMMUNICATIONS TECHNOLOGIES AND SERVICE ENGINEERING.

BARCELONA, JANUARY 2019

Dedication and acknowledgements

I would like to express my very great appreciation to Javier Ruiz, supervisor of this thesis, who has give me the opportunity to research in a project of this field in particular. Also for his constant and passionate thesis tracking and his valuable and constructive suggestions during the planning and development of this research work.

A special thanks to Eduard Alarcon, who provided to me new ways of approaching the research objectives and amount of tips to consider. His help was really useful for me. I am also deeply grateful with Napptilus Tech Labs, company with great interest in my research and who provided to me excellent motivation to ensure work continues along these lines.

I take this opportunity to express gratitude to all team of the Signal theory and communications department from my University, who provide to me their computational power of their machines to run my neural networks. Special thanks to Albert Gil, his help was really useful for me every time I needed to solve any problem.

Finally, I wish to thank my parents and friends for their support and encouragement throughout my study.

ABSTRACT

Home Staging concept is a set of techniques that allow to achieve more attractive aspect for the possible buyers of a building. In other words, the goal is to make the buyer feel at home when he tours the property. Home Staging is popular because it provides the buyer with an idea of how the furniture could be placed in the different spaces of the house and even, which is the best style according to the specific interests of each client.

The main idea on which this project is based, focuses on a tool concept with the purpose of give the opportunity to a possible client of a house, to preview how would be the empty rooms, bathroom or dining room with some type of virtual decoration, furniture and basic elements with a concrete style.

In particular, this project attempts to undertake a research of an specific part, which consist in analysis of changing the style of 2D images, specifically in bathroom images. So, using matching learning, deep learning techniques and other post-processing systems, we tried to modify the style (rustic, modern, asian, minimalist, etc.) of different bathroom images. This means that, without adding, deleting and even without changing the objects of the bathroom image, we are able to change the style of the furniture and walls.

In this document we will explain an approach using Convolutional Neural Networks to extract the style features from an image and replace them with another style. Moreover, improvements for this method and other systems are exposed.

Resum

El concepte de posada en escena d'una casa pertany a un conjunt de tècniques que permeten aconseguir un aspecte més atractiu per al possible comprador d'una casa. En altres paraules, l'objectiu és fer que el comprador se senti com a casa en el moment que estigui recorrent la propietat. La posada en escena de la casa, és popular perquè li brinda al comprador una idea de com es podrien col·locar els mobles en els diferents espais de la casa i fins i tot quin seria el millor estil segons els gustos específics de cada client.

La idea principal en què es basa aquest projecte, es centra en un concepte d'eina per tal de donar l'oportunitat a un possible client d'una casa, de previsualitzar com serien les habitacions buides, el bany o el menjador amb algun tipus de decoració virtual, mobiliari i elements bàsics amb un estil en concret.

En particular, aquest projecte intenta realitzar una investigació d'una part específica, consitent en l'anàlisi de l'estil de les imatges 2D, específicament en imatges de banys. Per tant, utilitzant l'aprenentatge automàtic i tècniques d'aprenentatge profund, intentem transferir l'estil (rústic, modern, asiàtic, minimalista, etc.) de diferents imatges de banys. Això significa que, sense afegir, eliminar i fins i tot sense canviar els objectes de la imatge del bany, som capaços de canviar l'estil dels mobles i les parets.

En aquest document explicarem un enfocament en particular de les xarxes neuronals convolucionals per a poder extreure les característiques d'estil d'una imatge i d'aquesta manera poder reemplaçar-la per un altre estil. Per altra banda, millores per a aquest mètode i altres sistemes, també estan exposats.

RESUMEN

El concepto de puesta en escena de una casa pretenece a un conjunto de técnicas que permiten lograr un aspecto más atractivo para el posible comprador de una casa. En otras palabras, el objetivo es hacer que el comprador se sienta como en casa cuando en el momento que esté recorriendo la propiedad. La puesta en escena de la casa, es popular porque le brinda al comprador una idea de cómo se podrían colocar los muebles en los diferentes espacios de la casa e incluso cuál es el mejor estilo según los gustos específicos de cada cliente.

La idea principal en la que se basa este proyecto, se centra en un concepto de herramienta con el fin de dar la oportunidad a un posible cliente de una casa, de cómo serían las habitaciones vacías, el baño o el comedor con algún tipo de decoración virtual, mobiliario y elementos básicos con un estilo concreto.

En particular, este proyecto intenta realizar una investigación de una parte especiífica, consistente en el análisis del estilo de las imágenes 2D, específicamente en imágenes de baños. Por lo tanto, utilizando el aprendizaje automático y técnicas de aprendizaje profundo, intentamos transferir el estilo (rústico, moderno, asiático, minimalista, etc.) de diferentes imágenes de baños. Esto significa que, sin agregar, eliminar e incluso sin cambiar los objetos de la imagen del baño, podemos cambiar el estilo de los muebles y las paredes.

En este documento explicaremos un enfoque de las red neuronales convolucionales para extraer las características de estilo de una imagen, para que de esta forma podamos reemplazarla por otro estilo. Por otra parte, mejoras para este método y otros sistemas, también están expuestos.

Table of Contents

Page

LIST OF TABLES

List of Figures

Chapter 1

Introduction

This project has been carried out at Univeritat Politecnica de Catalunya (Barcelona), department of Signal theory and communications, with the IMAGE group.

Home Staging using machine learning techniques is becoming increasingly popular as a consequence of the idea of preparing a property for sale could be a stressful and time-consuming endeavor. The main goal of this project, focuses on give the opportunity to a possible client of a house, to preview how would be all home spaces with some type of virtual decoration, furniture and basic elements in concrete style. With this approach, we could apply the principles and elements of design, but also how to appeal to the potential buyer.

There are two different ways to to deal with this Home Staging problem, either if the research is related with some kind of 2D image application or if the idea goes more deeply and the goal was to create a 3D house mapping. In addition, three different approaches must be considered:

- **Add furniture:** Making sure that the furniture is going to look good in a home is hard to imagine. Most of us are not equipped with the trained eye of an interior designer, which can lead to some regrettable purchases. That's where the investigation of Home Staging using Machine Learning techniques comes in handy. This system should allow customers to visualize either 2D or 3D of the virtual furniture in their homes. So, the sofas, tables, desks and chairs that you'd otherwise have to eyeball, can be virtually placed into a room, ultimately making the planning process much more accurate.

- **Delete furniture:** Furniture removal is an quite easy task to do manually, or semi manually (Photoshop, and even Power Point has such tools) if you use some kind of “marker” and edge detection. However, it’s a hard task to generate the background of the deleted object. Further, fully automated furniture removal plus background generation is quite a challenging task, and as far as we know, there is still no product that has satisfactory results with it, although some do try [3].

- **Style modification:** Furniture style describes the discriminative appearance characteristics of furniture. It plays an important role in real-world indoor decoration. Stylization is to combine the desired ‘style’ features and ‘content’ features separately extracted from different image references. Content means the shape and position of the objects in the image, while style is a set of unique characteristics such as coloring, patterns, textures.

All these three approaches have certain peculiarities that make them an a separate and specific research in which arise different problems and purposes to implement.

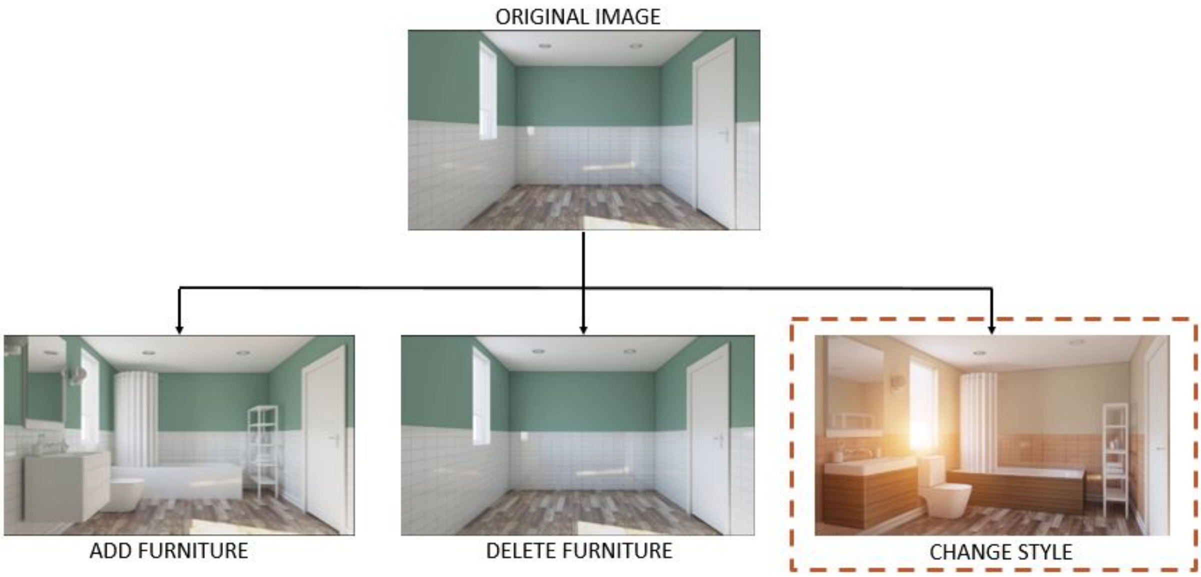


FIGURE 1.1. The global overview of the project consist of adding furniture, left of the figure, deleting furniture, center of figure and finally the stylization in the bottom right. This last one 'Change Style' is with which this project focuses.

In this project, we want to focus on the Style modification approach. In particular, transferring a specific style of an interior bathroom such as rustic, minimalist, classic, etc. onto a line drawing of another bathroom interior perspective.

Our approach is to use a deep neural network to extract the style features from the model image of a specific style to later be "transferred" into another one from a reference. Hence, with the content of any choice, the input image is generated by pixel-by-pixel continuous modifications until it's style becomes the same as the targeting style. In particular, Convolutional Neural Networks (CNN) are used to extract the style features from an image and transfer them to another.

The project main goals are:

1. Cover the field of Home Staging. Specifically, the style modification in terms of colours, textures and brightness of interior bathrooms.
2. Implementation of an algorithm based on Convolutional Neural Networks (CNN) to be able to modify the style of a bathroom image.
3. Obtain 2D image results with high quality using post-process systems responsible for improve the result details of the output image.

CHAPTER 2

STATE OF THE ART OF THE TECHNOLOGY USED OR APPLIED IN THIS THESIS:

Digital image processing deals with image manipulation to transform and perform some operations that covers many possibilities. This techniques has caused a great impact in the current systems we are using for visualization, restoration, classification, recognition, segmentation and stylization in images.

In this section, we will provide a brief summary of the actual investigations and technology used for style and texture extraction. Some of them, are useful systems to approach our objectives and as a consequence explained in our methodology chapter of this project. It starts with some citations of synthesization approaches in object appearances making use of basic techniques. Following, it is presented the current revolutionary technology that is in a deep research phase for it's successful outcomes, these are Machine and Deep Learning techniques. Finally, there are presented some researches for style and texture feature extraction using this revolutionary technology announced above. In addition, improvement algorithms to get better results are also detailed.

2.1 Style and Texture synthesis using basic techniques

Based on [18] [4], which implements several approaches using basic image techniques, they tried to synthesize the object appearances for analysis and application purposes.

In [18], they are demonstrating that many textures in daily life are statistically invariant in terms of colors and gradients. Hence, assuming that such statistics can be influenced by illumination, deformation and orientation, they are able to achieve texture replacement, surface relighting, as well as geometry modification by processing and recombining this discomposed components from a texture photo.

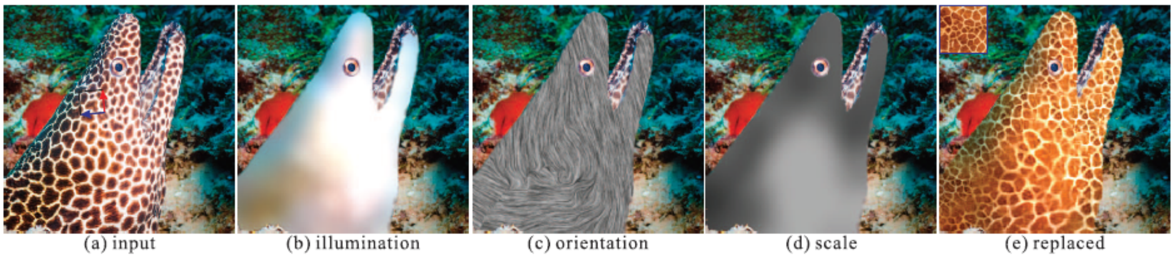


FIGURE 2.1. Image results of the technique proposed by [18]

On the other hand, technique [4] talks about an algorithm that renders an image with the desired appearance variations and synthesizes the output from a discrete set of annotated exemplars. They first segment the input data into a few clusters. Afterwards, they identify and assign weights to this clusters to finally apply the coarse-to-fine synthesis process that generates a candidate patch for each selected cluster and produces the result by merging the candidates.

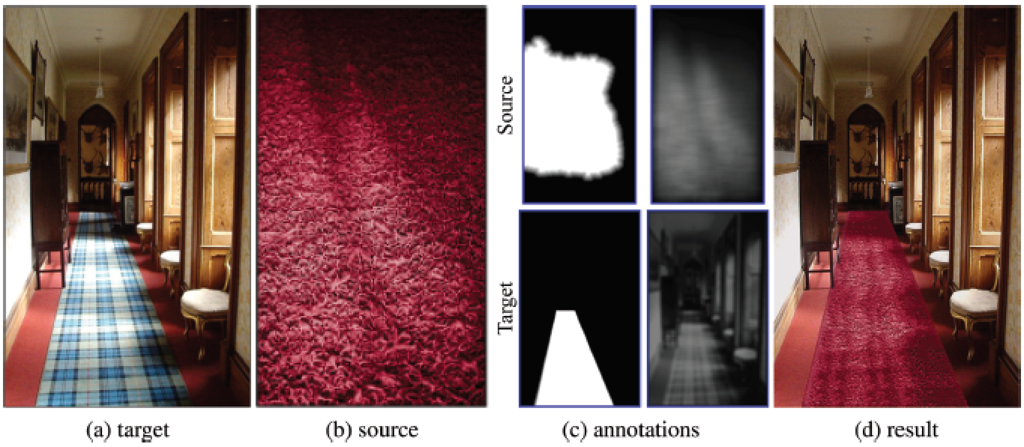


FIGURE 2.2. Image results of the technique proposed by [4]

They performs really well and achieves good results, but they don't have an optimized system that deals with the general properties of many styles or textures of the objects. Their approach is more focused only on a particular texture specialization.

2.2 Machine and Deep Learning techniques

Recently, machine learning and deep learning techniques are providing better implementations for this tasks. In particular, the most used class of deep neural network for image processing is Convolutional Neural Network's (CNN's).

CNNs are a subgroup of deep learning algorithms, which focus on computer vision and image processing tasks. These networks are loosely inspired by how the brain processes visual information and they are treated more like models with simple cells with local receptive fields, similar to filters or kernels, and complex cells, which are similar to pooling layers. In the field of convolutional neural networks, we can see that there are some related literature [14][29].

There are some reasons about why deep networks have become successful. Nowadays, we have more computation power due to Moore's Law, especially found in today's GPUs. Secondly, more training data is available in this moment, and finally we have in our hands new novel and better algorithms. For all these reasons, researchers described new ways to train CNNs more efficiently that allowed deeper networks to be trained. Recently, the performance increased significantly for multiple image databases, approaching or even beating human performance and many other pattern-recognition tasks. In general, CNNs are very good at classifying objects from an image, but also extracting image features.

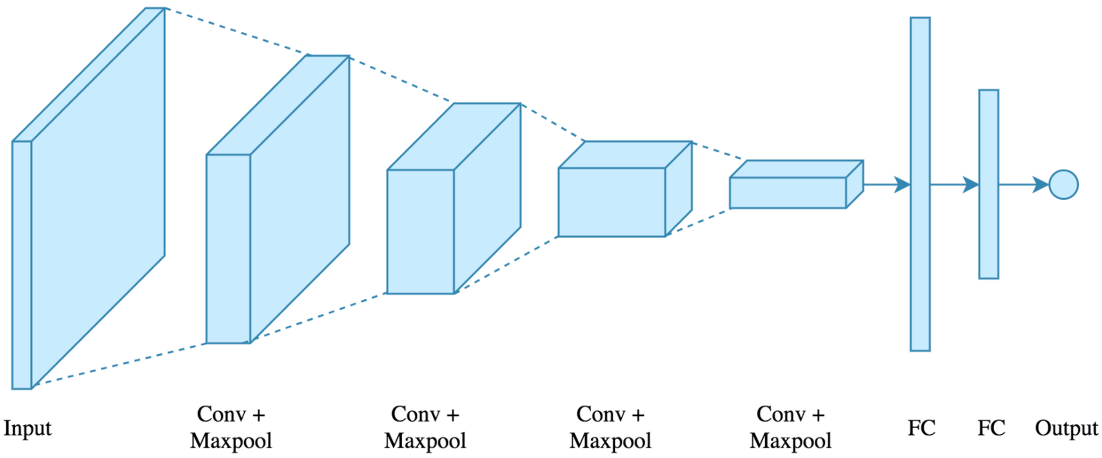


FIGURE 2.3. Outline of the main parts of a Convolutional Neural Network (CNN). The Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a back-propagation algorithm.

2.3 Stylization techniques with CNN's

Leon A. Gatys algorithm [5][6] describes the theoretical concepts about a Neural Network Algorithm for Artistic Style. This research, demonstrate that it's possible to separate and recombine the style of one artistic work from a painter and the content of another random image to synthesize it and imitate the style of the artistic work. In other words, his Style Transfer system is a process of redrawing an image in the style of another image while preserving the semantic content of the original image.

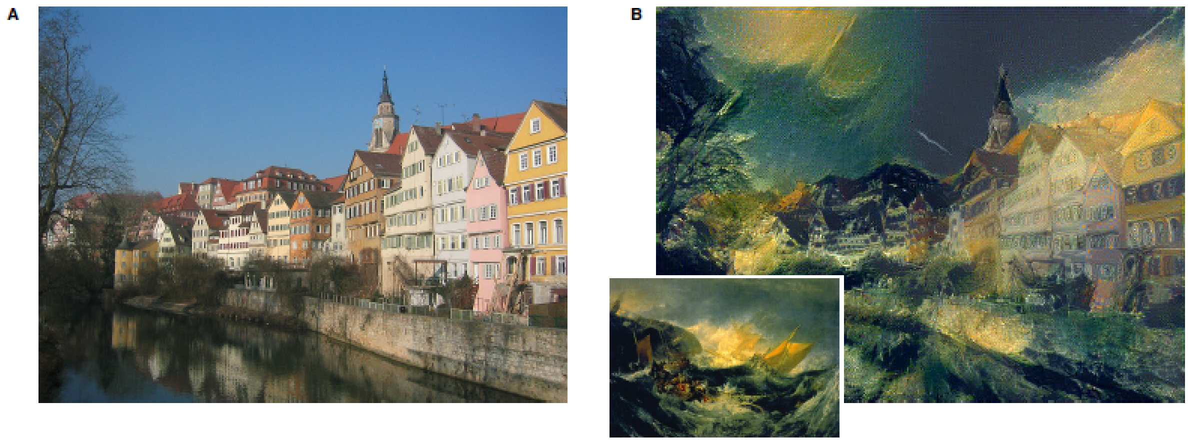


FIGURE 2.4. Artistic Style Transfer result from [6].

There are also other researches, which tried to improve different aspects of the system. Some examples as algorithm speed improvements[12][2], texture quality improve [27], systems to consider multi-reference image as style [1], [15] and even implementation of Style Transfer for video [23][9]. Our project start from the premise of this system, for this reason will detail the technical aspects in the methodology chapter. In addition, in this project, we also make use of other investigations. To summarize them, for image quality improvements, authors in [16] proposed a Smoothing step to once the output image from Style Transfer is obtained, the smoothing step ensures spatially consistent sylizations. On the other hand, to make use of not only one style image as a reference, [10] algorithm allow the possibility to Transfer the style from a data set of style images. Finally, authors in [24] propose methods for transfer style but using other networks, in particular Generative Neural Networks.

As said before, this lasts methods are technical explained in the methodology Chapter but focusing to our approach of stylize interior bathroom images.

Chapter 3

Methodology

In this chapter, we will detail the main methods used in this project. Beginning with the Style Transfer algorithm suggested by Gatys et al. [6], but immersed as a photo-realistic system to modify the particular style of an input bathroom image to another style. This method was a good starting point to focus our goals.

Once the Style Transfer system is developed and tested, we will realize in the results chapter that it creates some artifacts and distortions on the output image. For this reason, we will continue explaining which have been the improves to obtain better results using post processing. Especially, to increase the results quality, we took the post-processing idea from [16] where a smoothing filter is used to eliminate these artifacts. Then, we will evince that the Style Transfer method [6] doesn't use any style image data set to extract the style features and modify the input. Conversely, the system only takes a single image for style and content as a reference. That, could cause a generalization of the particular style meaning because of we are considering that a single image reference contain all the style properties. Accordingly to this, we will detail a modification of the Style Transfer algorithm based on [10] to analyze if considering the style features from a set of images belonging to the same style, provides consistent results.

Furthermore, in order to decrease time processing, compare and surpass the results of this multi-reference Style Transfer system, we researched about another method used with style data set reference which handle with a well known technique in Deep Learning named GAN's (Generative Adversarial Networks). Finally, we become aware that these systems implemented with large datasets of style images are not providing better results in comparison to the first Style Transfer method plus the photorealistic improvement. For this reason, we implemented a system to separately assign style in each particular bathroom object using the Style Transfer method plus photo-realistic system.

3.1 Style Transfer

Before explaining the system operation, it has to be clear that with this method we are not training any neural network. Principally, we are training specific values of an input image of the system. Hence, defining two distances, one for content D_C and one for style D_S, we are continuously transforming the input to minimize both, D_C with the content image and D_S with the style image to finally, obtain a representation of both.

3.1.1 CNN used

We have used the pre-trained VGG-19 Convolutional Neural Network [24] for the feature extraction. This net, consist of 19 layers deep trained with a wide range of images and this network was trained to perform object recognition either for a low or high level features. VGG-19 is quite famous because not only it works well, the creators also have made the structure and the weights of the trained network freely available online.

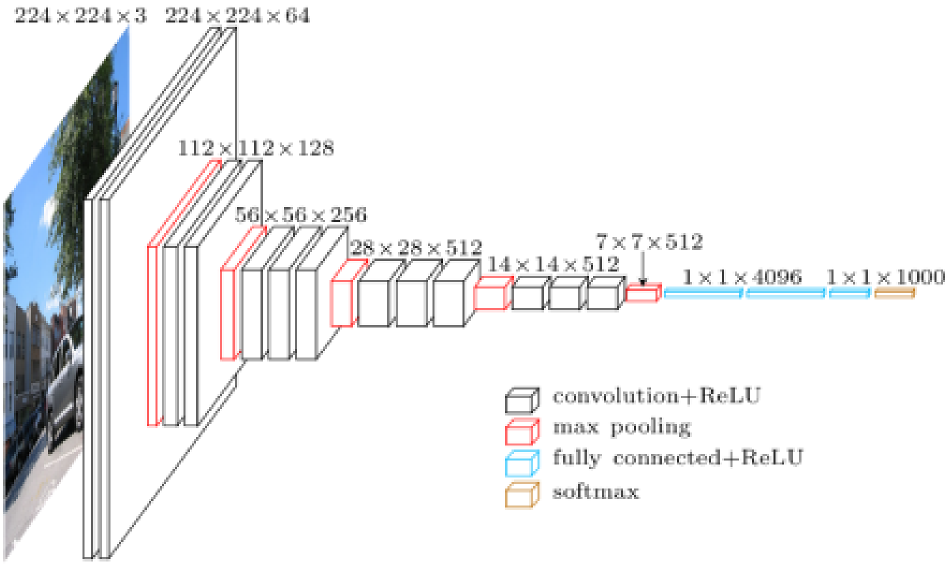


FIGURE 3.1. Network architecture of VGG-19 model. Figure reproduced from [24]

We use the VGG-19 neural network because we need the output features of each individual convolution layers to be able to measure content and style losses. This means that we do not use any of the fully connected layers of the original VGG-19 net, we only need the feature maps of each layer and not the final network label.

According to [24] the first layers of this network are extracting the small characteristics from an input image, contours, small changes, etc. as example. In contrarily, shapes, objects and big characteristics with the last layers. Now, we are in conditions to explain the Style Transfer methodology.

3.1.2 Content and Style extraction

The global structure of this system is displayed in *Figure 3.2* and the key of the Style Transfer method is using the encoding representation of each layer to calculate loss between the input image with respect to the content and style image. For this reason we divide the loss function into two parts, one is the Content loss and the other is the Style loss.

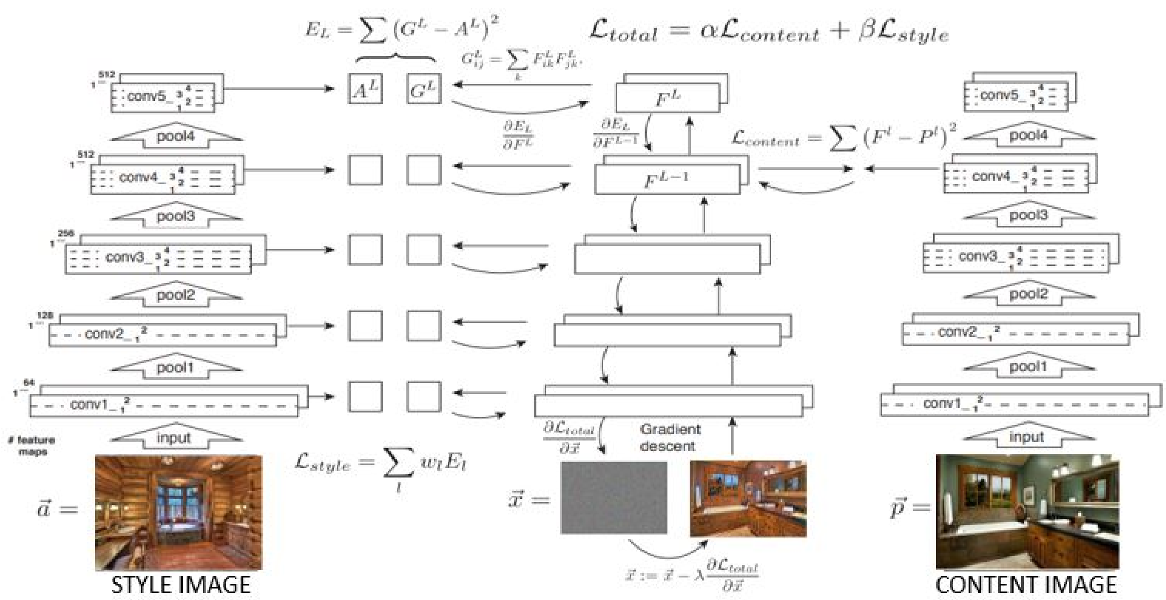


FIGURE 3.2. System overview. The algorithm begins with feature extraction A^l and P^l of the style image $\vec{a}$ and content image $\vec{p}$ respectively. Once we have the feature representations of both, an input image $\vec{x}$ (same content image in our case) is passed through the neural network in order to extract the style and content features G^l and F^l in the same way. In order to achieve the style loss L_{style} and content loss, $L_{content}$ we compute the mean squared difference between style and input feature and also content and input features. Immediately, the total loss L_{total} is computed linearly with both, providing weights α, β for each one. Finally, to be able to stylize the input image, the derivative of L_{total} (computed using error back-propagation) is repeatedly computer-processed to update the input image $\vec{x}$. Figure reproduced from [5]

LOSS FUNCTION

The combined style and content loss is a linear combination of the style and content losses where the α is the weighting of the content reconstruction and β is the weighting of the style reconstruction. These hiperparameters are useful for control how much of the content/style we want to inherit in the generated image.

$$L_{total}(\vec{a},\vec{p},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta L_{style}(\vec{a},\vec{x}) \tag{3.1}$$

CONTENT LOSS

The content loss is a function that represents a weighted version of the content distance (DC) for an individual layer. As we explained before, on the content image we want to capture the high-level content like shapes, objects and significant big features, for this reason, according to the VGG-19 net, it will be useful extract the map features of the higher layers. To calculate the content loss we pass both content image $\vec{p}$ and input image $\vec{x}$ through the network and get the feature representation of the 5th conv layer for both of these images. The distance is the mean square error between the two sets of feature maps:

$$L_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2 \tag{3.2}$$

Where **p** is the content image and P^l the feature representation in a layer **l**, **x** is the input image and F^l the respective feature representation in layer **l**.

STYLE LOSS

Following up on this basic idea, to extract the detailed characteristics of the style of an image, to capture texture information Gatys method [6] demonstrate that we can use a feature space, built on top of the filter responses in any layer of the network. This means that for the style loss we have to consider feature representations of many convolution layers from shallow to deeper layers of the model.

To be able to find the correlation between these feature maps across different filter responses in the same layer, we use the Gram Matrix. Gram matrix of a set of images (feature maps in our case) represents the similarity between two images. This matrix is the result of multiplying a given matrix by its transposed matrix:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{3.3}$$

Each element G(i,j) will represent the similarity measure between the feature map i and j. Now the style loss module looks almost exactly like the content loss module. The style distance is also computed using the mean square error between input and style gram matrix (G) and (A) respectively:

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G^l_{ij} - A^l_{ij})^2 \tag{3.4}$$

Finally the style loss function have the possibility to provide the contribution of each layer to the total loss:

$$L_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{3.5}$$

Hence, at the beginning of the algorithm, either with the content image in the right part or the style image in the left part, $\vec{p}$ and $\vec{a}$ are passed through the VGG19 network, filtered with the convolutional layers until the last layer. With this process we obtain the feature maps of each layer.

Subsequently, in each iteration the content (3.2) and style (3.5) losses are calculated as we detailed above, to finally calculate the total Loss function (3.1).

Moving forward, once we have both content and style loss we add them up to use an optimizer to perform gradient descent and modify the pixels values of the input image $\vec{x}$, such that it minimize the loss after each iteration. Then, we have used the L-BFGS algorithm to run our gradient descent and train the input image in order to minimize the content and style losses. We will repeat this procedure as many epochs we defined.

Finally we will obtain a reconstruction of the input image, but with aspects from the content image and with features from the style image.

3.2 Post processing

The Style Transfer method plus the modifications we applied to accomplish our objective is useful to obtain good enough image results as we could see in the Result Chapter 4.4. Therefore, some inconsistent artifacts and distortions are created by the method in the final output image.

These distortions caused for the Style Transfer method originally used for painting and artistic objectives makes the result less photo-realistic. Photo-realistism image stylization concerns transferring style of a reference photo to a content photo with the constraint that the stylized photo should remain photo-realistic.

We thought that some post-processing system should be applied to get rid of this significant distortions. Researching in systems to filter the noise of this images, we founded some ideas to deal with. For this reason, we take advantage of the post processing Photo-realistic Image Stylization [16] in which the proposed method consist on two steps: (1) Stylization and (2) Soothing step. Then, we focuses in this second step, where in our case, once we have the output image from Style Transfer, the smoothing step ensures spatially consistent sylizations.

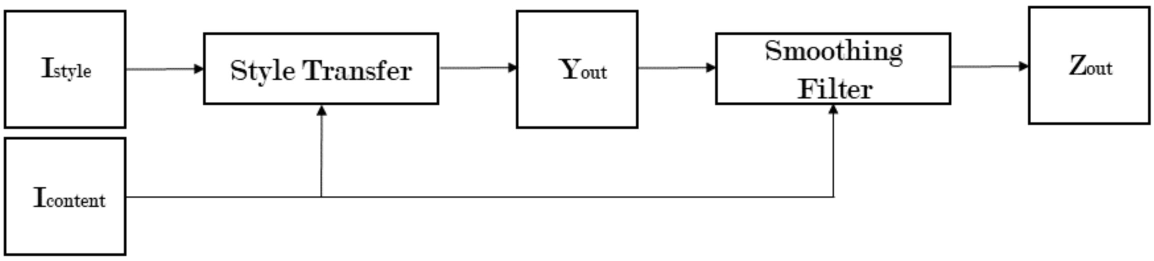


FIGURE 3.3. Scheme overview of Post-processing system. Y_{out} is obtained using the Style Transfer method, later this output is post processed giving Z_{out} as a result.

In the following subsection we will describe the filters and algorithms we used to clean the image from this pixel "issues".

3.2.1 Post processing System

The output result from the first Style Transfer method still looks non realistic because of the distortions of some part of the image. For this reason we aim to use this [16] post-processing method in which is defined photo-realistic smoothing part with two goals to consider:

- All neighbor pixels with similar content should be stylized similarly.
- In order to maintain the global stylization effects, the result of this system should not deviate significantly from the Style Transfer output.

To deal with this two goals, the system is trying to describe the similarities between pixels using an Affinity matrix [W=$w_{i,j}$] to be able to represent all pixels as a nodes in a graph. This matrix **W** is computed using the content photo and where the affinity between two pixels is based on means and variances of pixels in a local window.

As detailed in [16] the following optimization problem will model these two goals in which a smoothness term and a fitting term are defined:

$$\arg\min_{r} \frac{1}{2}\left(\sum_{i,j=1}^{N} w_{ij} \left\| \frac{r_i}{\sqrt{d_{ii}}} - \frac{r_j}{\sqrt{d_{jj}}} \right\|^2 + \lambda \sum_{i=1}^{N} \|r_i - y_i\|^2\right) \tag{3.6}$$

Where R is the optimal solution, this means R is the smoothed version of Y (the output from Style Transfer). Hence, y_i is the pixel color in the Y result and r_i is the pixel color in the desired smoothed output R.λ controls the balance of the two terms. This optimization problem is about to minimize the pixel color distance in the desired smoothed output R. Motivated with the graph-gased algorithms [22], in which the optimal solution R is based on the pairwise pixel affinities to be able to achieve a consistent stylization within semantically similar regions.

3.3 Multi-reference in Style Transfer

In this point, we reached the conclusion of transferring the specific style of an image should not come from one only style reference. In the previous sections, we were applying the Style Transfer method from Gatys [6], using only one image for style extraction. This means that we are only taking the style features from the reference image we assign as a style. Therefore, there are some other general factors, textures and style coloration that can't be obtained from only one image. For this reason, we thought about improving the Style Transfer method, trying to add some system to consider not only one style image, but considering a large data set of style image from the same style.

3.3.1 Multi Reference method

In this section we will focus in [10] algorithm, this paper propose an Style Transfer algorithm which transfer the general features of certain style to generate a stylization of an input image with more than one style image reference.

System structure

The system is structured in two parts: (1) Model creation of the desired style. (2) Estimate the style and content with the input image, to obtain a stylized output.
At the beginning, the system develop a representation of the desired style from a bathroom data set of style images. Afterwards, the Style Transfer model from Gatys [6] is used to estimate an appropriate texture to render the input image.

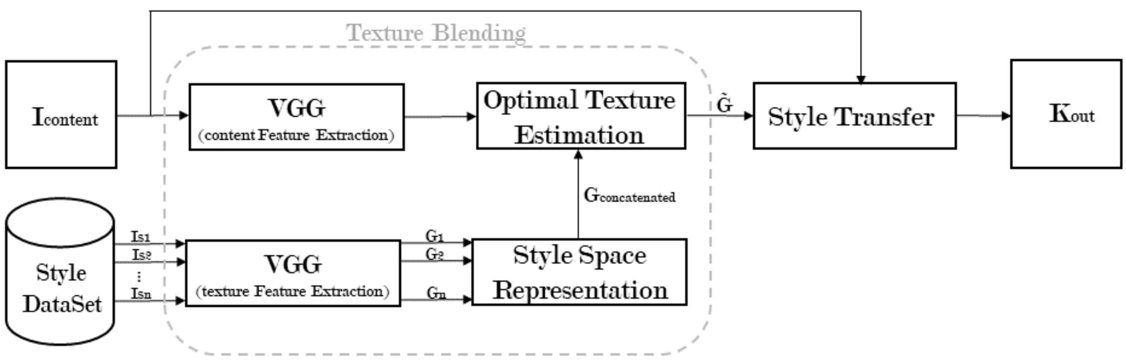


FIGURE 3.4. Scheme overview of the multi-reference system. First, making use of the feature maps G_1,G_2...G_n from the Style Dataset images, a style space representation is created $G_{concatenated}$. Subsequently, the system optimize and estimate the final texture features $\tilde{G}$ to finally apply the Style transfer method.

In *Figure 3.4* is detailed overview of this multi-reference system where to create the general model of the desired style (1), the paper is based in the idea of a normalized version of the style feature space **G** used, where **G** is the Gramm matrix. On the other hand, the second part (2) is the same operation as Style Transfer method, explained above

Then, it is shown that the style representation **G** of a concatenated image is approximately equal to the linear combinations of the G of each of the concatenated images. In addition, G of different images belonging to the same style can be linearly blended to represent the general texture.

$$G^l_{ij}(I_1 \circ I_2 \circ ... \circ I_N) = \sum_{n=1}^{N} \frac{Ml,n}{Ml,all} G^l_{ij}(I_n). \tag{3.7}$$

Where $G_{i,j}(I_n)$ is a summation of the filter responses, $M_{l,n}$ is the number of pixels for each filter response for the n_{th} image, and $M_l, all = \sum_n M_{l,n}$

Once we have obtained the linear combination of G of the style images, if we consider content image as an input image $I_{input} = I_{content}$, the algorithm is searching the feature map element of each style image which approximates to the feature map element of the input image.

$$\arg\min_r \sum_{l,i,j} \left(G^l_{ij}(\mathbf{I}_{content}) - \sum_{n=1}^{N} r_n G^l_{ij}(\mathbf{I}_n) \right)^2 \tag{3.8}$$

So, for each element **j**, of the response of the **i** filter in the layer **l** of the feature maps of the style images, the algorithm is trying to find the nearest element regarding the input image.

This $r = [r_1, ..., r_n]$ is the blending ratio of the texture images. So, we construct a linearly weighted texture feature,

$$\tilde{G}^l_{ij} = \sum_{k=1}^{K} r_k G^l_{ij}(\mathbf{I}_k) \tag{3.9}$$

Finally the Gatys method is applied:

$$L_{total} = \alpha \sum_{i,j} \frac{1}{2N_l} \left(F^l_{ij}(\mathbf{I}) - F^l_{ij}(\mathbf{I}_{content}) \right)^2 + \beta \sum_l \sum_{i,j} w_l \left(G^l_{ij}(\mathbf{I}) - \tilde{G}^l_{ij} \right)^2 \tag{3.10}$$

3.4 GAN method

As we have been seen in the last multi-reference with Style Transfer method, the complexity of the system increase significantly and consequently the time spent to extract the output is being affected. For this reason, we have resorted to apply another method as an alternative to achieve deacrese the processing time.

In this section, we will implement the system provided by [30] in which they obtain quality results changing the style domain of an image to another style domain. This approach is totally different with the Style Transfer method we've been explaining until now, but with same goal: Transfer a particular style of an image to another style.

Their method uses a type of neural network currently new and with impressive results. The **GAN's** (Generative Adversarial Networks) are a revolutionary class of algorithm used to generate photographs that look at least superficially photo-realistic. In a GAN setup, there are two differentiable functions, represented by neural networks. The generator and the discriminator, both with different roles:

- The generator trying to maximize the probability of making the discriminator mistakes its inputs as real.
- The discriminator guiding the generator to produce more realistic images.

In the perfect equilibrium, the generator should capture the general training data distribution. As a result, the discriminator would be always unsure of whether its inputs are real or not.

3.4.1 Algorithm operation

The method proposed by [30] tries to capture the special features of a image data set to subsequently translate it from a source domain X to a target domain Y. So, if $G : X \rightarrow Y$, their goal using first a training step is to learn mapping functions between this two domains, such that the distribution of images from *G(x)* is indistinguishable from the distribution Y using an adversarial loss.
They adopt an *Adversarial Loss* to learn the mapping such that the translated images cannot be distinguished from images in the target domain. Furthermore, the learned mapping functions should be cycle-consistent because of adversarial losses alone cannot guarantee that the learned function can map an individual input x_i to a desired output y_i. So, to further reduce the space of possible mapping functions, *Cycle Consistency Loss* is also implemented.

To equalize the generated image distribution towards the distribution of the destination domain, two adversarial discriminators are defined:

- D_X: Tries to distinguish between images from X domain and translated images $F(y)$
- D_Y: Tries to distinguish between images from Y domain and translated images $F(x)$

Then, they express the objective as:

$$L_{GAN}(G,D_Y,X,Y) = E_y[logD_y(y)] + E_x[log(1-D_Y(G(x))] \tag{3.11}$$

As it's been said, Adversarial training can learn mappings G and F that produce outputs identically distributed as target domains Y and X respectively.

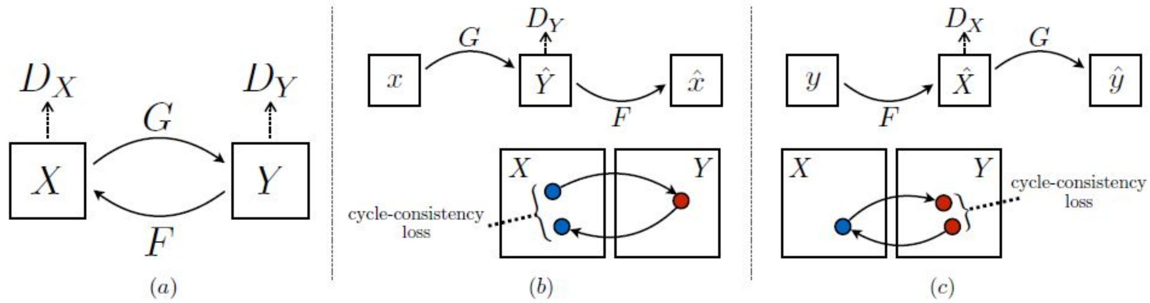


FIGURE 3.5. Two different domains are reproduced. After image translation, Cycle-consistency should be able to bring back each image x from domain X to the same domain. Figure from [24]

But, we also have to ensure that the learning mapping functions should be cycle-consistent. Hence, for each image x from domain X, the image translation cycle should be able to bring x back to the original image, and they implemented the *cycle consistency loss*:

$$L_{cyn}(G,F) = E_x[\|F(G(x))-x\|_1] + E_y[\|G(F(y))-y\|_1] \tag{3.12}$$

Finally, applying this two loss jointly, we aim to achieve this following loss function:

$$L(G,F,D_X,D_Y) = L_{GAN}(G,D_Y,X,Y) + L_{GAN}(G,D_X,Y,X) + \lambda L_{cyn}(G,F) \tag{3.13}$$

Then, the algorithm aims to minimize the G and F maps to reduce the error generating an image for the other domain, and maximize the D_X and D_Y to improve the ditinction between the translated samples and real samples:

$$G*,F* = \arg\max_{G,F} \arg\min_{D_X,D_Y} L(G,F,D_X,D_Y) \tag{3.14}$$

3.5 Segmentation

As we will see in Results chapter, in our subjective point of view, these lasts two methods (Multi-reference with Style Transfer) and (GAN's) in which we were using an a database as a style reference, we didn't achieve as many realistic results in compassion to the basic Style Transfer optimized with the post-processing photo-realist system.

Therefore, we have to remind that the problem with this method was using only one image as a style reference. And this led us to research in other methods as we have explained above.

Now, we know the best results are obtained with the Style Transfer plus photo-realist system. But, at this moment, we were applying stylization to the whole image at the same time, this means that we are changing the style of all the bathroom furniture belonging to the image.

In this section, we go more deeply with this method, we tried to focus in changing the style of each object of the image independently with different style reference. Thinking about how this would fit in the project, we believe this would be useful for someone who only wants to change the style of an specific object. In this way, it is possible to pre-view how would be this object with the desired style. Thus, this will be a second post-process system after implement the photo-realistic filter in the Style Transfer output.

FIGURE 3.6. General scheme with two post-processing systems. Smoothing Filter and Segmentation respectively.

For this approach, we need to create some masks for each region where we want to apply an a concrete style transfer. We should use segmentation system to segment each particular object, but this is a hard task which is out of scope of this project. For this reason we take advantage of one editing tool, Photoshop to manually extract binary masks of our image.

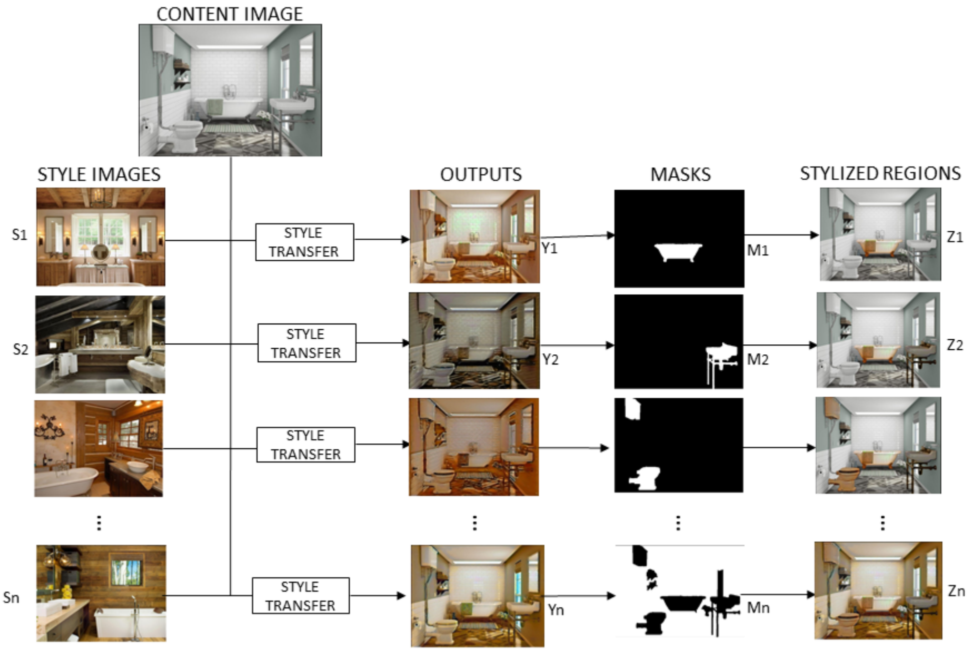


FIGURE 3.7. Style Transfer in specific region of the content image. First, different outputs are extracted $(Y_1, Y_2, ..., Y_n)$ applying Style Transfer + Photo-realism system using a set of Style image $(S1, S_2, ..., S_n)$ belonging to the same style. Then, masks $(M_1, M_2..., M_n)$ are applied in order to stylize the desired region. White pixels referenced as stylized output, black pixels as a content background.

On 3.7, the procedure is detailed, where the same input content image is firstly computer-processed with the Style Transfer algorithm with different reference images of same style. Secondly each output is post-processed with the photo-realist filter to finally apply the binart masks to assign an output style y1, y2, .. to each region in particular.

Chapter 4

Results

In this chapter, we want to present the results obtained using the explained methods in methodology chapter. But before starting with it, we will describe the Data Base used for train and test our Neural Networks. Afterward, we will detail the specifications of an a classifier trained with the objective of distinguish between bathroom styles. This classifier has been useful to create another way to measure our results apart from our subjective measure. Finally, we will show the results of the stylization methods with some test images to evaluate them subjectively and in addition, the chapter will be complemented with tables of results of more tests using an objective measure, our classifier.

4.1 Data Set

It's a hard task to find cleaned and accurate bathroom Data Set labeled with each different style. For this reason we used the steps described in [25] to be able to generate our own bathroom data set from Google images. Using the JavaScript console from Google browser and small Python code we achieved big amount of images with automatic download of the current search in Google images we were interested. The Google browser allow us to specify considerably search characteristics apart from text, as *Size, Type, Resolution, etc.*, useful for precise image specifications.

We investigate which styles of bathroom existed, we found: *Rustic, Minimalist, Classic, Asian, Modern, Country, Industrial, Mediterranean, Colonial, Tropical, Eclectic, Scandinavian, etc*. Using [25] procedure, we have generated a small data set of the 3 most common bathroom styles to test the stylization methods: **Minimalist, Rustic, Classic**.

(a) Minimalist (b) Rustic (c) Classic

FIGURE 4.1. Images that belong to our Data Set according to the style. Minimalist style is shown in **a)**, rustic in **b)** and finally classic style in **c)**.

On the other hand, there has been some problems using a Data Base from Google images:

- Some of these images were advertising.
- There were some bathroom images with the company watermark
- Google does not have good labeling and gives you some image of certain style that you or a interior designer would say it is different style.
- Image limitation. We were able to just download 600 images of each style.

To make our Data Base more confident, we made a purge of all data sets, checking each image and deleting the advertising and wrong style image. Therefore, with this action, we reduced the amount of images of each style.

Style	**Number images**	**Resolution**
Minimalist	485	1024 x 768 pixels
Rustic	360	1024 x 768 pixels
Classic	511	1024 x 768 pixels

TABLE 4.1. The specifications of our dataset

4.2 Classifier

Texture and style, is often a significant component for the visualization of an interior design of a bathroom and not all the people is able to determine which is the style with accuracy like a designer do. Image classification is a challenging task, traditionally approached by computational models. To analyze, process and finally classify the input image of the model, image acquisition, management, communication, and processing systems are important to be able to identify patterns and characteristics of each style.

The goal was then, develop a classification system to acquire an objective measure to distinguish between bathroom styles apart from the subjective measure that each of us could do. With this, we were be able to test a large number of results and check which class of style belongs to.

4.2.1 Classifier training

Taking advantage of our bathroom Data Set we had obtained, we constructed a Data Base with 'Train' and 'Evaluation' Data Set for each specific style separated with 63% for training process and 37% for evaluation process:

	Training	Validation	
Rustic	210	150	**360**
Minim	309	176	**485**
Classic	330	181	**511**
	849	**507**	

TABLE 4.2. Number of images for train and validate the classifier.

With this image data set, we expected to obtain a classifier with high quality, using Deep Learning Techniques. As [13][26] we could train an entire Classification Neural Network from scratch, but other researches in this field, advise that we can't obtain good enough accuracy with this reduced Databases like ours with 1000 bathroom images in total. In practice, very few people train an entire Convolutional Network from scratch because it is relatively rare to have a dataset of sufficient size.

Instead, exist a technique called Transfer Learning [7] where a first pre-trained network is used, to then re-train the learned features, or transfer them, to a second target network to be trained on a target data set and task. This process tend to work if the features are general, that is, suitable to both base and target tasks, instead of being specific to the base task. It is common to pre-train a Convolutional Netetwork on a very large data set (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the Net either as an initialization or a fixed feature extractor for the task of interest.

There are some methods to do this:

- **Fine-tuning the ConvNet**:This method initialize all weights with the pre-trained model values. After this, we modify the output layer (fully-connected layers) with the output labels we want. At this point, the Net replace and retrain all layers.

- **ConvNet (Convolutional Network) as fixed feature extractor**:This other method, fix all the weighs of the network an only replace and retrain the final fully-connected layers.

We applied the first method **Fine-tuning the ConvNet** taking advantadge of the pre-trained Network **ResNet-18** [8], in our particular case the best in terms of accuracy.

	Classic	Minimalist	Rustic
Classic	**119**	51	11
Minimalist	41	**124**	11
Rustic	31	21	**98**

TABLE 4.3. Confusion matrix.

In Table 4.3 we can see the performance of style classification model. This confusion matrix notice that the performance of this classifier between Classic and Minimalist bathroom style is not good at all. But making several training tests, we achieve as a best classifier results with an Accuracy of : **67%** in validation.

We release that there are different factors which the Accuracy results can vary and these are some factors we think as most important:

- Quality of the training and validation images plays an important role. Our Data Base of interior bathroom images could cause some impact to the classifier training.

- Parameters of the network: number of epochs, step size, learning rate..

With this Accuracy, we considered this classifier as a indicative measure of the style of our bathroom results, but without giving all the validity to this classification.

In the Appendix chapter there are some examples where this classifier assign style percentages according to the input image.

4.3 Results

In this section, we will report the results obtained with the stylization methods using machine learning techniques and post-processing systems.

First, we will report fine-tuning results with the basic Style Transfer method and evaluated with the classifier. The parameters values, such as the learning rate, momentum, weight decay were chosen first referencing to the training parameters described in the respectively papers of the model [6], and then in *Figure 4.3* by taking advantage of the classifier percentage results, we tried to find the optimal hyper-parameters as Number of epochs and content/style relationship to train.

Subsequently, we will show the improves of using the post-processing method. So, taking the output images from the Style Transfer outlined above, in *Figure 4.4 and 4.6* we represent some examples of images treated with the photo-realistic filter.

Next, in *Figure 4.6* we will demonstrate how multi-reference image for the Style Transfer method is suffering improvements or deterioration of the output results according to the number of images for the style data set we are using.

At the same subsection, we present the last method we worked on, the GAN's method (Generative Adversarial Networks). Otherwise, we will use the same content image and same style Data Set to compare the output results between Multi-Reference Style Transfer method and GAN's method. In such a way, in *Figure 4.8* we will be able to distinguish the results between them.

We become aware that these last two methods using style Data Base are not overcoming the Style Transfer method with photorealistic filter. So, finally, we take this best method to implement it with segmentation masks. So, in *Figure 4.9* taking manual segmented masks, we will see results of local style transfer, which divides the image into multiple regions, usually furniture, to obtain the style of each region.

4.3.1 Style Transfer and Post-Processing

Style Transfer can manipulate the representations of content and style independently to produce new stylized images. In our experiments the images are synthesized by matching the content representation on layer 'conv4_2' and style on layers 'conv1_1', 'conv2_1', 'conv3_1', conv4_1' and 'conv5_1'. But, this method have some other parameters that influence significantly to the results, in particular the number of epochs we use to process the input and the relation content between style (α/β) respectively.

We take an analysis of this parameters in order to get the optimal values. In *Table 4.2* we tried to test the same reference image, but evaluating each possibility using the classifier. In particular we tested the images from *Figure 4.2*, in which is presented the optimal result.

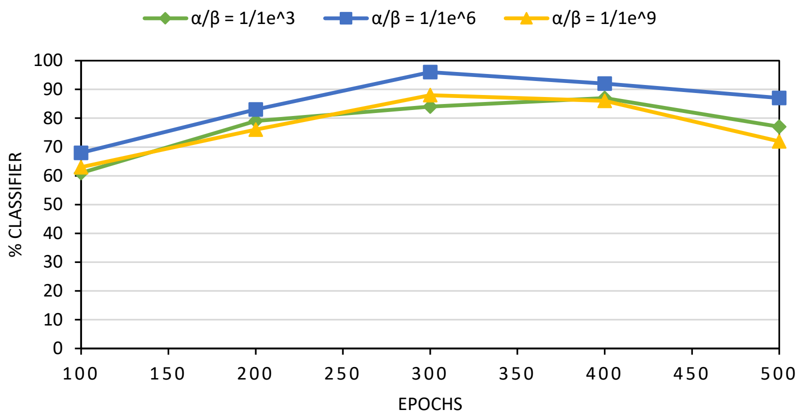
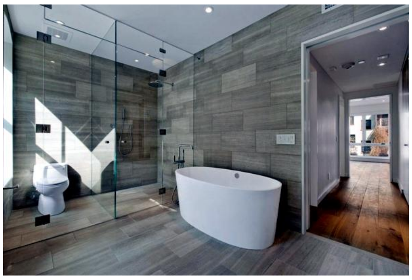


FIGURE 4.2. Best parameters for Style Transfer according to the classifier: 300 epochs and relation $\frac{\alpha}{\beta} = \frac{1}{1000000}$.



(a) *input*

(b) *style*

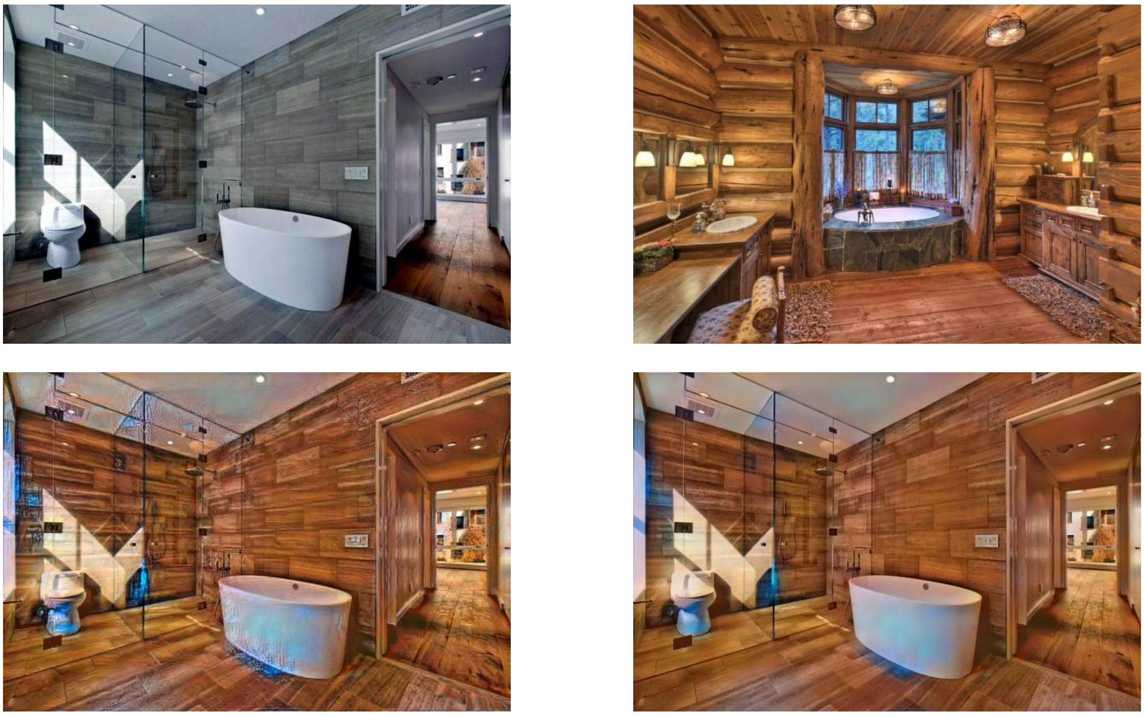
(c) *output*

FIGURE 4.3. Using the optimal parameters according to the classifier result, we show one example of the output result (c) of the Style Transfer method. Where (a) Content image, (b) Style image and finally (c) Output stylized. The parameters used are: 300 epochs and relation $\frac{\alpha}{\beta} = \frac{1}{1000000}$.

As explained in (Section 3.2), if we check the output image (c) from *Figure 4.3* we can observe that some distortions and artifacts are created. At this moment, in order to get rid of this distortions and solve this issue, we decided to make use of a post-processing system. In the following *Figures 4.4 and 4.6* we will show some bathroom examples of different styles applying this filters.

(a) CONTENT - 95.8% minimalist

(b) STYLE - 98.4% rustic



(c) ST OUTPUT - 98.3% rustic

(d) FILTERED OTUPUT - 92.0% rustic

FIGURE 4.4. Style Transfer and Photo-realist filter results. Transferring Style from minimalist to rustic style and measuring the classification percentage of each case.

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	minimalist	minimalist	0.7	95.8	3.5
(b)	rustic	rustic	98.4	0.2	1.4
(c)	rustic	rustic	98.3	1.5	0.2
(d)	rustic	rustic	92.0	7.6	0.4

TABLE 4.4. Percentage of style classification for each image.

(a) CONTENT - 69.1% rustic

(b) STYLE - 90.7% minimalist

(c) STOUTPUT - 70.3% minimalist

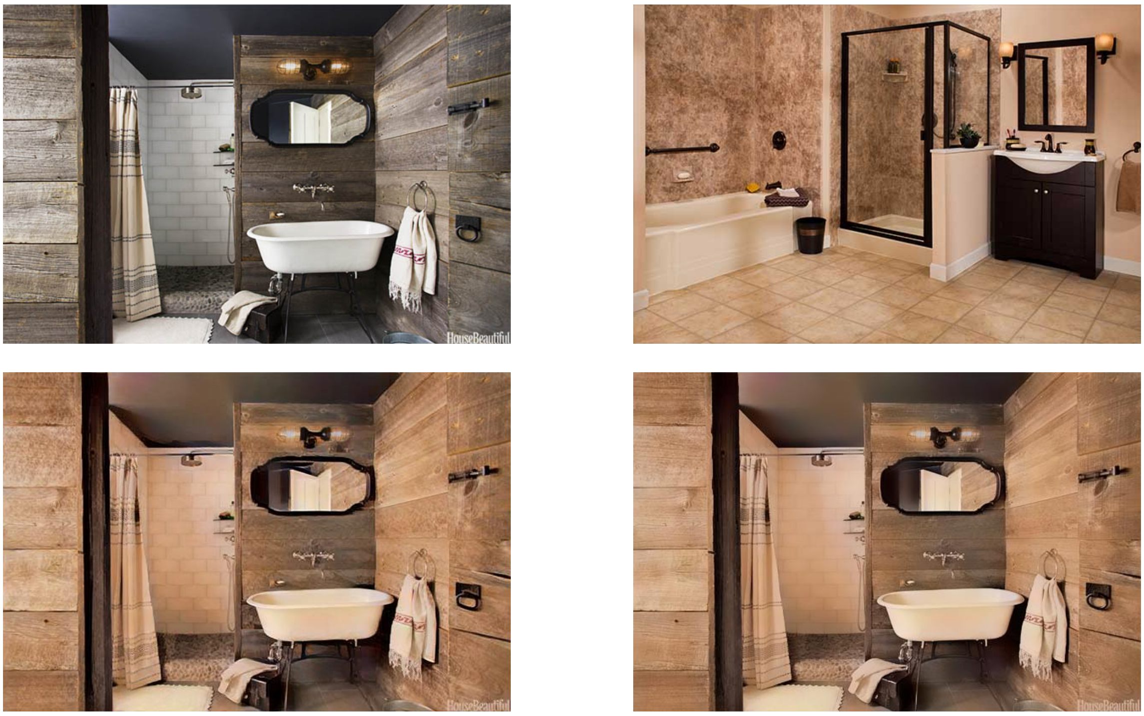
(d) FILTERED OUTPUT - 72.8% minimalist

FIGURE 4.5. Style Transfer and Photo-realist filter results. From rustic to minimalist style. Measuring the classification percentage of each case.

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	rustic	rustic	69.1	29.7	1.2
(b)	minimalist	minimalist	5.0	90.7	4.3
(c)	minimalist	minimalist	6.9	70.3	22.8
(d)	minimalist	minimalist	19.0	72.8	8.2

TABLE 4.5. Percentage of style classification for each image.

(a) CONTENT - 65.5% rustic (b) STYLE - 55.7% classic



(c) OUTPUT - 51.8% classic (d) FILTERED OUTPUT - 55.8% classic

FIGURE 4.6. Style Transfer and Photo-realist filter result from rustic to classic style. Measuring the classification percentage of each case.

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	rustic	rustic	65.5	7.4	27.1
(b)	classic	classic	7.5	36.7	55.7
(c)	classic	rustic	41.2	7.2	51.6
(d)	classic	classic	7.5	36.8	55.8

TABLE 4.6. Percentage of style classification for each image.

4.3.2 Multi-Reference Style Transfer and GAN's

Moving on to the methods which use more than one image as a style reference, we first show the results based on multi-reference Style Transfer method. In our tests we realized that depending of the number of style images we take as reference, the output is more or less stylized. As said in (Section 3.3), we consider that the main reasons of this problem are due to the inconsistent Data Base. In the algorithm, we are trying to take the closet feature maps between the input and all the style Data Base. So, if some style image is similarly to the content image (same style), the system is forcing to change as little as possible. Hence, as many style image we have as a reference, less stylization is being transferred.

(a) content

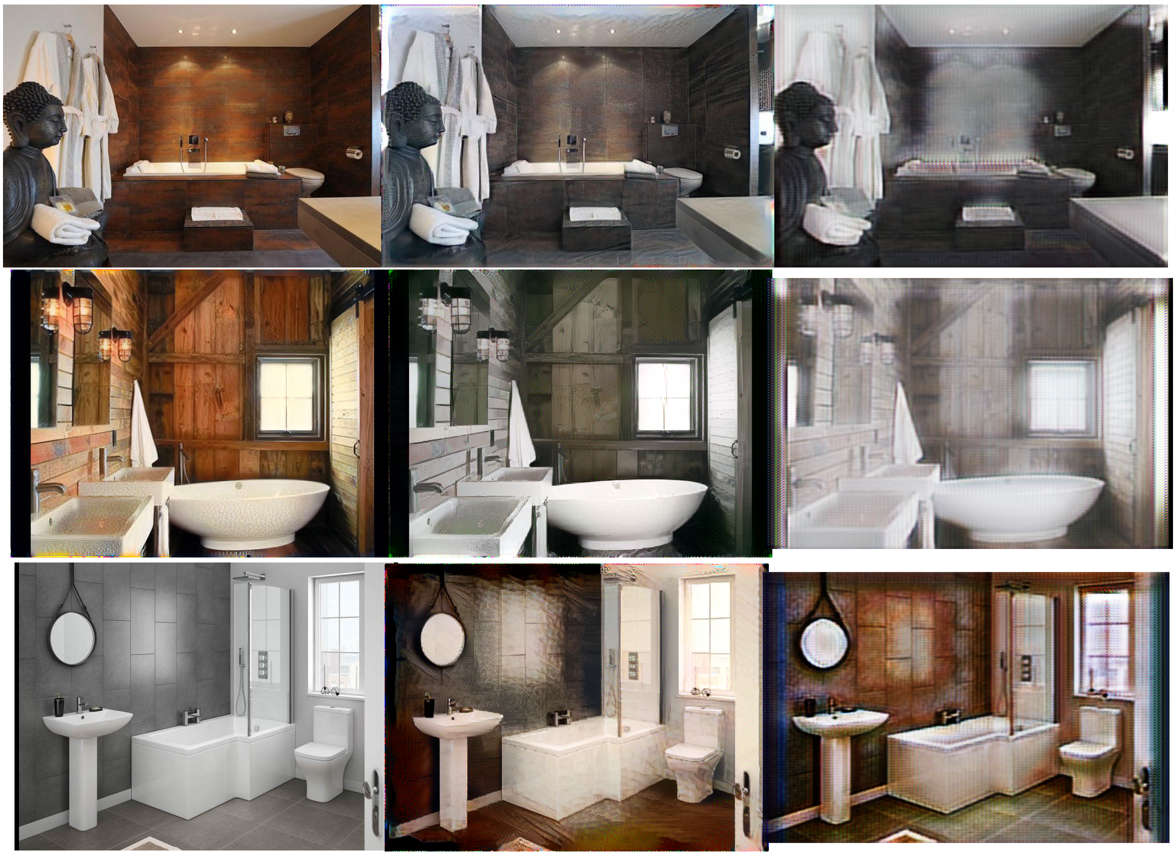
(b) DB = 10img

(c) DB = 30img

(d) DB = 100img

FIGURE 4.7. The Multiple reference method using Style Transfer is causing stylization decay as many reference we have. In (c) 30 image reference we obtain the best result in our subjective point of view.

GAN's method also performs the stylization of an image from an style DataBase. In *Figure 4.8*, using the same reference image we compare the multireference Style Transfer method (*Section 3.3*) and GAN's method (*Section 3.4*) across a series of indoor bathroom scenes.



(a) Inputs (b) Muti-reference results (c) GAN results

FIGURE 4.8. Qualitative comparisons between *Multi-Reference Style Transfer* and *GAN* methods.

Time is also an important parameter that can give us an idea of how much time is spent in processing the image. For the neural networks we have tested, we also checked in *4.7* the time spended in stylization.

Method	**Time Training**	**Time Transfer**
Style Transfer	N/A	15s
ST + photorealistic	N/A	19s
Multiple - Reference	N/A	1min 12s
GAN's	51min	8s

TABLE 4.7. Time comparison of different methods

In relation to this aspect, we have to take account of the power computation we are using, time also depend of the characteristics of our gpu/s of the computer or server. Our experiments, have been measured taking advantage of the power of our University servers: GPU=16G, RAM=20G.

4.3.3 Segmentation

The segmentation of the stylizated image was carried out in order to change the style of each object/furniture of the bathroom image independently with different style reference. In *Figure 4.9* we show an example with binary masks used to segment each object and subsequently be stylized.

(a) Content

(b) Style Image (c) Output (d) Mask (e) Segmented Result

FIGURE 4.9. Stylization process for the desired regions using binary masks.

CHAPTER 5

Conclusions and future development

As we told in the introduction, Home Staging is becoming increasingly popular and takes high importance when somebody wants to visualize the interior design of rooms, bathrom, etc. Traditionally, the people who want to take over a new home didn't have the possibility to pre-view the interior design. Furthermore, there either have the option to preview the current furniture and design but with other distribution and style.

Our research wants to cover one of the parts to deal with the Staging Home problem. In particular, the goals are the implementation of an algorithm to extract features and modify the style of a bathroom image using Machine Learning techniques based in Convolutional Neural Network systems to obtain good enough results with high quality.

The Style Transfer method purposed by Gatys [6] would provide a point of departure for our objective, following with post process systems to get better photo-realist results. In particular, these has been finally the best results we have obtained in terms of quality and timing. Additionally, we have investigated new methods to balance out the principal inconsistency of the Style Transfer method. Hence, using an image database as a style reference, we tried to implement another systems to subsequently visualize the output results.

With regard to the systems implemented in this project, we can assure that this research has provide an idea of which level can reach these systems focusing only in the modification of texture and color of the input image. Hence, using the visual subjective measure of each person, we have been able to show results that approach photo-realism. In addition, using a classifier, trained by us, we have been able to give an objective measure to our results, providing which percentage of style belong to.

Finally, we can ensure that our final system presents a considerable improvement in performance, although adding more post processing steps also increases the computational cost and the time of operation.

At this point, we have in our hands the possibility of implementing these systems as a tool to change textures of the interior parts of a house, but it would be necessary change the perspective of this research and focus ourselves in other systems that not only modify the style or texture but also create the elements and position of them.

Bibliography

[1] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, *Stylebank: An explicit representation for neural image style transfer*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 2770–2779.

[2] T. Q. Chen and M. Schmidt, *Fast patch-based style transfer of arbitrary style*, CoRR, abs/1612.04337 (2016).

[3] Y.-L. Chen and C.-T. Hsu, *Towards deep style transfer: A content-aware perspective*, in BMVC, 2016.

[4] O. Diamanti, C. Barnes, S. Paris, E. Shechtman, and O. Sorkine-Hornung, *Synthesis of complex image appearance from limited exemplars*, ACM Trans. Graph., 34 (2015), pp. 22:1–22:14.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge, *A neural algorithm of artistic style*, CoRR, abs/1508.06576 (2015).

[6] ——, *Image style transfer using convolutional neural networks*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), pp. 2414–2423.

[7] L. G. Hafemann, L. E. S. de Oliveira, P. R. Cavalin, and R. Sabourin, *Transfer learning between texture classification tasks using convolutional neural networks*, 2015 International Joint Conference on Neural Networks (IJCNN), (2015), pp. 1–7.

[8] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), pp. 770–778.

[9] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, *Real-time neural style transfer for videos*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 7044–7052.

[10] H. Ikuta, K. Ogaki, and Y. Odagiri, *Blending texture features from multiple reference images for style transfer*, in SIGGRAPH Asia Technical Briefs, 2016.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 5967–5976.

[12] J. Johnson, A. Alahi, and L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, in ECCV, 2016.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in NIPS, 2012.

[14] Y. LeCun, L. Bottou, and P. Haffner, *Gradient-based learning applied to document recognition*, 1998.

[15] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, *Diversified texture synthesis with feed-forward networks*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 266–274.

[16] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, *A closed-form solution to photorealistic image stylization*, in ECCV, 2018.

[17] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, *Visual attribute transfer through deep image analogy*, ACM Trans. Graph., 36 (2017), pp. 120:1–120:15.

[18] X. Liu, L. Jiang, T.-T. Wong, and C.-W. Fu, *Statistical invariance for texture synthesis*, IEEE transactions on visualization and computer graphics, 18 (2012).

[19] F. Luan, S. Paris, E. Shechtman, and K. Bala, *Deep photo style transfer*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 6997–7005.

[20] ——, *Deep painterly harmonization*, Comput. Graph. Forum, 37 (2018), pp. 95–106.

[21] R. Mechrez, E. Shechtman, and L. Zelnik-Manor, *Photorealistic style transfer with screened poisson equation*, CoRR, abs/1709.09828 (2017).

[22] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, *Color transfer between images*, IEEE Computer Graphics and Applications, 21 (2001), pp. 34–41.

[23] M. Ruder, A. Dosovitskiy, and T. Brox, *Artistic style transfer for videos and spherical images*, International Journal of Computer Vision, (2018), pp. 1–21.

[24] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR, abs/1409.1556 (2014).

[25] D. M. Sollami, *google image search hack*, 2017.

[26] C. SZEGEDY, W. LIU, Y. JIA, P. SERMANET, S. E. REED, D. ANGUELOV, D. ERHAN, V. VANHOUCKE, AND A. RABINOVICH, *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015), pp. 1–9.

[27] D. ULYANOV, A. VEDALDI, AND V. S. LEMPITSKY, *Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 4105–4113.

[28] T.-C. WANG, M.-Y. LIU, J.-Y. ZHU, A. TAO, J. KAUTZ, AND B. CATANZARO, *High-resolution image synthesis and semantic manipulation with conditional gans*, in CVPR, 2018.

[29] M. D. ZEILER AND R. FERGUS, *Visualizing and understanding convolutional networks*, in ECCV, 2014.

[30] J.-Y. ZHU, T. PARK, P. ISOLA, AND A. A. EFROS, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, 2017 IEEE International Conference on Computer Vision (ICCV), (2017), pp. 2242–2251.

Appendix A

In this thesis report, we have seen some stylized bathroom examples using different methods, and details about the accuracy of our classifier. But, in this section, we will add some more results and also some examples of the classifier detection:

Classifier


(a)


(b)


(c)

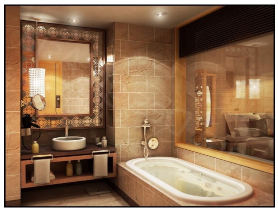

(d)

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	classic	minimalist	1.2	57.6	41.2
(b)	classic	classic	6.7	12.3	81.0
(c)	classic	classic	4.4	8.1	87.5
(d)	classic	classic	3.7	3.4	92.9


(a)


(b)


(c)

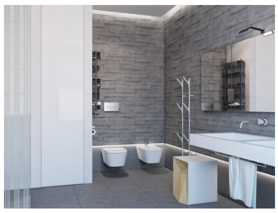

(d)

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	minimalist	minimalist	0.2	98.8	1.0
(b)	minimalist	minimalist	1.4	95.7	2.9
(c)	minimalist	minimalist	0.5	99.1	0.4
(d)	minimalist	minimalist	2.7	92.9	4.4

(a) (b) (c) (d)

Image	True Style	Predicted Style	Rustic %	Minimalist %	Classic %
(a)	rustic	rustic	81.5	3.2	15.3
(b)	rustic	rustic	97.2	0.7	2.1
(c)	rustic	rustic	98.8	0.4	0.8
(d)	rustic	rustic	98.1	0.5	1.4

Stylization results

(a) CONTENT (b) STYLE

(c) ST OUTPUT (d) FILTERED OTUPUT

(a) CONTENT

(b) STYLE

(c) ST OUTPUT

(d) FILTERED OTUPUT