Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Kartografie und Geoinformation
Institute of Cartography and Geoinformation

Rushan Wang

If Van Gogh or Picasso Painted Maps Explorations of Neural Style Transfer Applied to Maps

HS22 Semester Project

Institute for Cartography and Geoinformation
ETH Zurich

Supervision

Sidi Wu
Prashanth Chandran
Dr. Magnus Heitzler
Prof. Dr. Lorenz Hurni

January 2023

Acknowledgements

I am deeply grateful to Professor Lorenz Hurni for providing me with the opportunity to work on this interesting topic as my semester project. I would also like to extend my sincere thanks to Sidi Wu and Prashanth Chandran for their invaluable support and guidance throughout the course of this research. In addition, I am grateful to Dr. Magnus Heitzler for offering his valuable advice during the development of this project. Their expertise and support were crucial in helping me to complete this work successfully.

Contents

Abstract

Neural style transfer is a technique for stylizing images by transferring the visual style of a reference image to a target image while preserving the content of the target image. Developing effective and efficient methods for style transfer for maps could improve the aesthetic appeal and functionality of topological maps in a variety of applications. In this project, we aimed to create a more comprehensive method for transferring map style. We explored the application of neural style transfer to maps, with a focus on stylizing topological maps (Siegfried map). We compared several state-of-the-art style transfer methods, including AdaIN and AdaConv, and evaluated the performance of different style loss functions and model architectures. To improve the readability and clarity of maps, we introduced a contrast loss term that aims to enhance the contrast between different feature classes in the map. To evaluate the effectiveness of our approach, we conducted a user study to gather feedback on the results of our style transfer process. Our experiments showed that AdaConv was the most effective method for preserving the style structure of the input image, while gram loss was the most effective style loss function. We also found that general models, which were trained on a diverse range of styles and contents, tended to perform better in preserving the overall style structure of the input image. These results suggest that neural style transfer can be used to enhance the visual appeal and clarity of topological maps and that there is potential for further improvement by fine-tuning the model architecture and training process.

Chapter 1

Introduction

Expressive cartographic renderings are useful for all mapmakers, and in the context of personalized map design, more sophisticated tools could improve the design process and the resulting maps [3]. Deep learning techniques can be applied to maps to assist with map stylization.

Existing studies on map stylization tend to use Generative Adversarial Networks (GANs) techniques [18, 12], which is a type of deep learning model used to generate new, synthetic data similar to a training dataset. GANs are not designed to preserve the content of a specific input image, but rather to learn the overall distribution of the training data and generate new samples that are similar to it. While GANs can be used to generate images that contain certain types of content, they do not have the ability to specifically preserve the content of a given input image. Therefore, if the goal is to preserve the content of a specific input image while transferring the style of another image onto it, neural style transfer may be a better approach than using a GAN.

Neural style transfer is a method of combining the content of one image with the style of another image using deep learning. The resulting image preserves the content of the original image but has the style of the second image. The style includes the color, stroke, and texture of the image, which can be extracted as features from the image thanks to convolutional neural networks(CNNs). This can be used to create a variety of interesting and creative effects, such as transferring the style of a famous painting onto a photograph or combining the content of a photograph with the style of a drawing. Some studies [7, 14] also show that style transfer can be used for data augmentations for improving model robustness. Neural style transfer is implemented using a convolutional neural network (CNN) trained to optimize a loss function that combines the content and style losses of the images. The content loss measures how different the content of the generated image is from the content of the original image, while the style loss measures how different the style of the generated image is from the style of the second image. By minimizing these losses, the neural network is able to generate an image that preserves the content of the original image while taking on the style of the second image.

In the field of neural style transfer, many studies have focused on changing the style of the photo-realistic image to the style of paintings. In this study, we want to explore the neural style transfer applied to maps. Due to the unique characteristics of maps, challenges would occur in this task.

One challenge of applying neural style transfer to maps is that maps often contain a lot of detailed, complex information that can be difficult to preserve when transferring the style of an artistic image onto a map. This is because the neural style transfer process involves modifying the pixel values of the input image in order to achieve the desired style, which can lead to the loss of some of the finer details and nuances of the original map. Additionally, maps often have a specific visual structure and layout that is important for conveying information clearly and accurately, and this can be disrupted by the neural style transfer process.

Another challenge is that maps often have a very specific and uniform visual style that is designed to be clear and easy to read, and it can be difficult to achieve a visually appealing result when transferring the style of another image onto a map. This is because the style of the second image may not necessarily be well-suited to the content and structure of the map, which can lead to a final image that is confusing or difficult to interpret.

In this work, we aim to explore the use of neural style transfer for maps. Experiments were conducted with various model settings and compared state-of-the-art methods in neural style transfer, AdaIN [9] and AdaConv [1]. To improve the readability of the maps, we also proposed the use of contrast loss to enhance contrast among different feature classes. Finally, we carried out a user study to evaluate our results and facilitate future research on this problem.

The structure for this report is as follows. In Chapter 2, we will introduce recent studies on the related topic. In Chapter 3, our methodology for this project will be presented. The results of the study will be presented in Chapter 4, followed by a discussion of these findings in Chapter 5. Finally, Chapter 6 will provide the conclusion of the work.

Chapter 2

Related Work

2.1 Neural Style Transfer

2.1.1 Overview

Neural style transfer was first introduced by Gatys *et al.* [6]. It allows users to render content images with the style of other artistic images. Dumoulin *et al.* [5] demonstrated that with a deep network artistic style of diverse paintings can be captured. Their methods enabled the transfer of any style between images, but it was based on a slow optimization process [1]. Ulyanov *et al.* [16] speed up the style transfer method by proposing a feed-forward neural network that is style-specific and pre-trained.

Apart from using convolutional neural networks, some approaches [19] have used adversarial networks to regularize the generation of stylized images by learning the intrinsic properties of image styles from large-scale multi-domain artistic images.

2.1.2 Instance normalization

The pioneering work of Ioffe and Szegedy [10] introduced the use of the batch normalization layer, which greatly improves the training of feed-forward networks by normalizing feature statistics.

Instance normalization [16] is a technique used in deep learning to normalize the activations of a layer for each sample in a batch independently, rather than normalizing the activations across the entire batch. This is done by subtracting the mean of the activations for each sample and dividing by the standard deviation. The resulting normalized activations have zero mean and unit variance for each sample. Instance normalization layer is found to be effective in feed-forward style transfer [17].

Inspired by the instance normalization layer, Huang [9] introduced Adaptive Instance Normalization(AdaIN), which adjusts the mean and variance of the content input to match those of the style input. By transfer feature statistics, AdaIN shows good performance in arbitrary style transfer.

However, AdaIN only modifies global statistics and is insensitive to localize, spatial semantics in the style input. To address this issue, Chandran *et al.* [1] proposed Adaptive Convolutions(AdaConv), an extension of AdaIN, which allows for the capture and transfer of both the global statistics and local structure of the style.

To gain a clearer understanding of AdaIN and AdaConv, the following are the equations for them. Given the feature of content input x and the feature of style input y, let μ and σ be the mean and

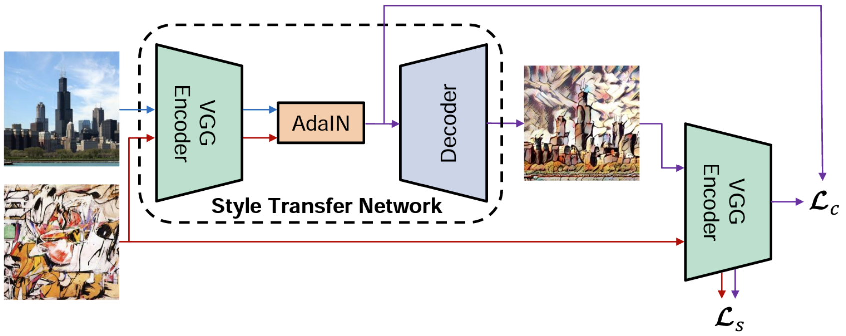


Figure 2.1: An overview of style transfer algorithm with AdaIN[9]. Content image and style image are fed into the encoder network to get the content feature and style feature. The style feature is then used to normalize the content feature, through a process of scaling and shifting in the AdaIN section. The normalized content feature is then passed through the decoder network to generate the stylized image.

standard deviation over the feature channel,

$$AdaIN(x, y) = \sigma(y)(\frac{x - \mu(x)}{\sigma(x)}) + \mu(y) \tag{2.1}$$

AdaConv extends AdaIN by first introducing a conditioning 2D style filter f, which allows for spatially varying modulation for the feature channel. Given $N(x)$ as the neighborhood around sample x,

$$AdaConv_{dw}(x, f) = \sum_{x_i \epsilon N(x)} AdaIN(x, f_i) \tag{2.2}$$

Then it turns the convolution into a depthwise separable convolution by including a separable pointwise convolution tensor $p\epsilon\mathbb{R}^c$, which helps it more effectively model the correlation between different input channels c,

$$AdaConv(x, p, f) = \sum_c p_c AdaConv_{dw}(x, f_c) \tag{2.3}$$

2.2 Map style transfer

Map stylization has traditionally been achieved using non-photorealistic techniques such as painting, drawing, technical illustration, and animated cartoons [4]. These techniques involve specifying style parameters and related non-photorealistic rendering techniques to achieve an expressive look.

Map style transfer involves identifying the key characteristics of the desired style and reproducing them through the composition and arrangement of visual elements [4]. Several previous studies on map style transfer have employed Generative Adversarial Networks techniques, including Pix2Pix [11] and CycleGAN [19] models. Kang *et al.* [18] presented a framework that utilizes GANs to transfer pre-existing style criteria to multiscale maps, without the need for input from CartoCSS map style configuration sheets. Christophe *et al.* [4] conducted experiments to evaluate the use of Pix2Pix and CycleGAN models on ortho-images, using a range of map designs for different geographic locations, from simple styles (such as plan maps) to complex styles. Other studies focus on rendering satellite images to various styles of maps. MapGAN [13] was proposed as a means of generating accurate, multi-type electronic maps quickly, using both remote sensing images and render matrices. SMAPGAN [2] was proposed as a semi-supervised generation of styled map tiles directly from satellite images

There has been a lack of research on rendering topological maps, which differs from synthesis maps generated from satellite images in that they should prioritize preserving map content. Additionally, when using GANs to generate maps from satellite images, the target stylized map is in the same domain as the transferred data, both are geospatial data. However, in the case of stylizing topological maps to a Van-Gogh style map, the desired image may not lie within any specific domain, making it difficult to find appropriate training data.

Our objective is to create a more comprehensive method for transferring map style, rather than specifying individual styles for each geometric element and trying to optimize the overall style composition. By doing so, we hope to improve the visual appeal and usability of topological maps in various applications.

Chapter 3

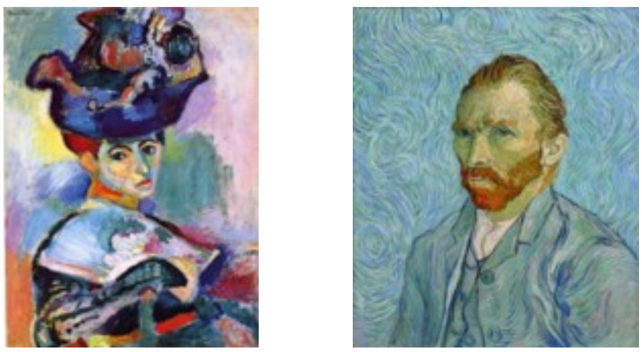
Method

3.1 Data

The Topographic Atlas of Switzerland, also known as the Siegfried Map, is a comprehensive Swiss national map series that was published between 1872 and 1949 [8]. In our work, the original map was cropped to the size of (3,512,512) and used as content images, the three channels are R, G, and B channels. Style images were selected and downloaded from Kaggle [1]. Figure 3.1 shows some examples of the content image and style image used in this project. Figure 3.3 shows the legend of the maps.

(a) Content 1 (b) Content 2 (c) Content 3

Figure 3.1: Example of Content Images.



(a) Style Fauvism (b) Style Van Gogh

Figure 3.2: Example of style Images.

[1]https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time

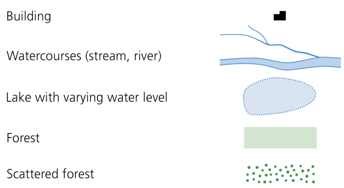


Figure 3.3: Legend of Maps.

For the semantic mask of the map, as shown in Figure 3.4, we manually drew the boundaries of each feature class. In our work, we mainly focus on 8 classes: lake, river, wetland, road, building, forest, vegetation, and background. However, due to the significant overlap between different feature classes in the map, we had to include the overlap area in both feature masks when dealing with overlap between two different feature classes.

(a) Forest mask (b) Lake mask (c) Vegetation mask (d) Wetland mask

Figure 3.4: Example of Semantic mask for Content 1.

3.2 Network Overview

The overall architecture of the model adapted from AdaConv [1] is shown as Figure 3.5.

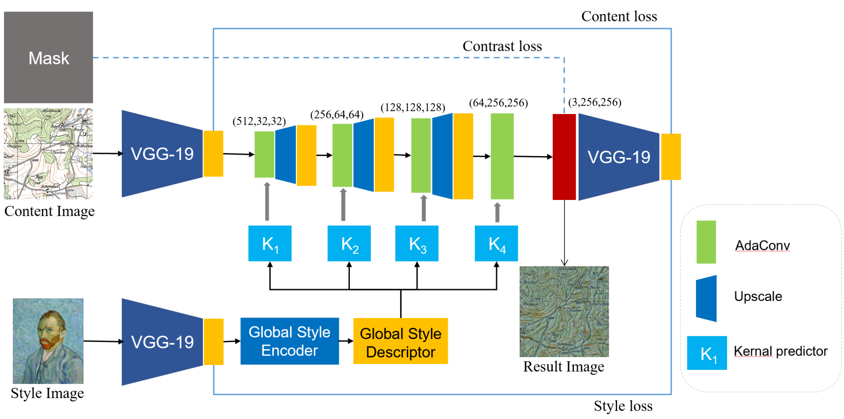


Figure 3.5: Network overview with AdaConv[1]. The VGG-19 encoder network extract style feature and content feature. The style feature is fed into the global style encoder to get a global style predictor, which is then used to predict four kernels. The predicted kernels are applied to the content feature during the decoder network to get the stylized image.

The pre-trained VGG-19 network is able to extract meaningful features from images [15]. This makes it a useful tool for tasks such as style transfer, where we can use the features extracted by the VGG-19 network to manipulate the style of an image.

In this architecture, the input content image will be cropped to (3,256,256) and then passed through the pre-trained VGG-19 network. The output from the $relu4\_1$ layer of the VGG-19 network, known as the content feature, will be used to calculate the content loss later.

These procedures are the same for the style image to get the style feature. The style feature is then fed into the global style encoder to get the global style descriptor. From the global style descriptor, the kernel predictors(K_1, K_2, K_3, K_4) would predict depthwise separable convolutional kernels with pre-channel biases [1]. The kernels are applied to all corresponding layers in the decoder, and the final output is the style transfer result.

In order to compute the content loss and style loss, the output result should also be fed into the pre-trained VGG-19 network.

The mask in Figure 3.5 is a binary image with 8 channels, where each channel corresponds to a mask for a specific map feature class.

3.3 Global Style Encoder

In a deep network, the receptive field refers to the region in the input space that can influence the features of a particular layer. In this work, the global style encoder was applied to make the receptive field cover the whole style image, which consists of several convolution layers, average pooling operations, and leaky ReLU activation. Moreover, a fully connected layer was added at the end to make the receptive field spans all input. The output style feature was then reshaped to match the size of the kernel predictor.

3.4 Kernel Predictor

Kernel predictor aims to generate depthwise separable kernel based on the global style descriptor. The depthwise separable kernel can significantly reduce the number of parameters and computations in a model, while still allowing it to capture rich, hierarchical representations of the input data. Generally, it consists of two separate parts: a depthwise kernel and a pointwise kernel. The depthwise kernel operates independently on each channel of the input, performing a convolution across just the spatial dimensions (i.e. height and width) of the input. This allows the kernel to learn different spatial filters for each input channel, allowing the model to capture more fine-grained spatial features. The pointwise kernel operates on the output of the depthwise kernel, performing a convolution across all channels of the inputï¼ which allows the model to learn a set of cross channel interactions between the different feature maps learned by the depthwise kernel.

In our case, each kernel predictor K_n output spatial kernels, pointwise kernels, and a bias term, which then were applied to modulate input content features.

3.5 Training

Pre-trained VGG was used to compute the loss function to train the decoder. The loss function includes three parts, content loss, style loss, and contrast loss,

$$L = L_{content} + \alpha L_{style} + \beta L_{contrast} \tag{3.1}$$

3.5.1 Content Loss

The content loss is calculated to measure the difference between the feature representations of the transferred image and the original content image, which allows us to quantify how well the transferred image preserves the content of the original image. In this work, the content loss was calculated as the mean squared error between the content features lc_i and the features of the output image lo_i,

$$L_{content} = \sum_{i=1}^{D} (lc_i - lo_i)^2 \tag{3.2}$$

3.5.2 Style Loss

The style loss is for measuring the difference between the style of the transferred image and the original style image, allowing us to quantify how well the transferred image captures the style of the original image. We tried two different style loss functions, namely moment matching loss and gram loss [6].

Moment Matching Loss

Moment matching loss encourages the output of a model to match the statistical moments of a target distribution. In our case, the mean and variance of the style features ls_i and output features are calculated,

$$L_{style(mm)} = \sum_{i=1}^{D} (\mu(ls_i) - \mu(lo_i))^2 + \sum_{i=1}^{D} (\sigma(ls_i) - \sigma(lo_i))^2 \tag{3.3}$$

Gram Loss

Gram loss[6] is a widespread method to measure style loss in the field of image style transfer. It is based on the idea that the style of an image can be captured by the correlations between the different channels of its feature maps, which can be represented as the gram matrix of the feature maps.

The gram matrix G^l_{ij} is the inner product between the vectorized feature map i and j in layer l. For gram matrix of style feature layer G_{ls} and gram matrix of output feature layer G_{lo},

$$G_{ls} = \sum_{i,j} ls_{i,j} \times ls_{j,i} \tag{3.4}$$

$$G_{lo} = \sum_{i,j} lo_{i,j} \times lo_{j,i} \tag{3.5}$$

By comparing the gram matrices of the feature maps of two images, it is possible to measure the difference in style between the two images,

$$L_{style(gram)} = \sum_{i=1}^{D} (G_{ls} - G_{lo})^2 \tag{3.6}$$

3.5.3 Contrast Loss

We incorporated a contrast loss component to improve the distinction between different feature classes within the map. This was achieved by maximizing the dissimilarity among different feature classes and minimizing the dissimilarity within the same feature class. The dissimilarity was calculated by measuring the distance of the color distribution in the color space. Standard deviation and cosine distance were used as two different approaches. Given N is the number of feature classes, and f_n is the cluster of the points within the feature classes n.

For the standard deviation method,

$$L_{contrast} = \sum_{n=1}^{N} std(f_n) - std(f_1, f_2, ..., f_N) \tag{3.7}$$

For cosine distance method, the equation is the same, except we change the calculate of standard deviation to cosine similarity.

Chapter 4

Result

4.1 Comparison

All experiments in this work were conducted in the same environment: Visual Studio Code, python 3.10.4, and CUDA 11.7. The default iteration is set to 160000. And the default learning rate is 0.0001.

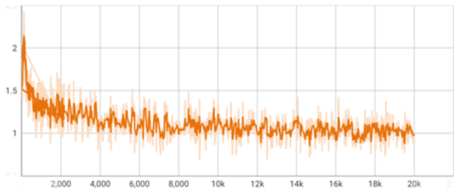


Figure 4.1: Example of content loss.

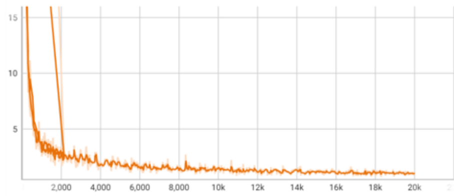
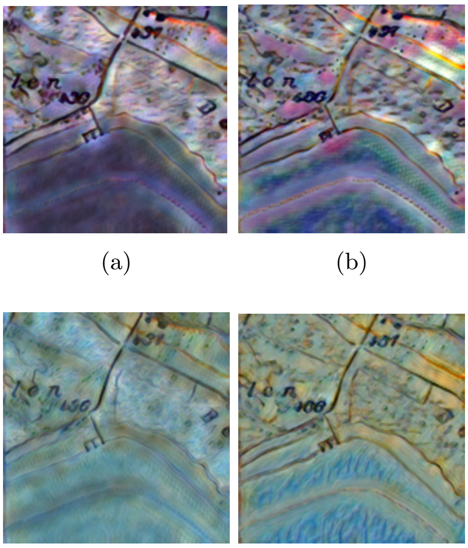


Figure 4.2: Example of style loss.

4.1.1 Effect of Network Architecture

In Figure 4.3, the results of the network with AdaIN and the network with AdaConv are compared. Figure 4.3a and Figure 4.3b are stylized in Fauvism. Figure 4.3c and Figure 4.3d are stylized in Fauvism. It is notable that results from AdaConv have better performance in preserving the structure from the style image. Results from AdaIN only maintain the global statistics of the style image. Therefore, in the following experiments, we only focused on the network with AdaConv.



(a) (b)

(c) (d)

Figure 4.3: Effect of AdaIN and AdaConv,(a)(c) are results from AdaIN, (b)(d) are results from AdaConv.

4.1.2 Effect of Style Loss Function

In Figure 4.4 the style transfer results of employing moment matching loss and gram loss are compared. Figure 4.4a is stylized in Fauvism and with moment matching loss. Figure 4.4b is stylized in Fauvism and with gram loss. Figure 4.4c is stylized in Van Gogh and with moment matching loss. Figure 4.4d is stylized in Van Gogh and with gram loss. It is obvious that with gram loss, the results have better texture and restore the color distribution in the image better. However, the improved style performance of gram loss appears to come at the cost of destroying the original line segment features of the image. For instance, the contour line in Figure 4.4d is broken.

4.1.3 Effect of training set

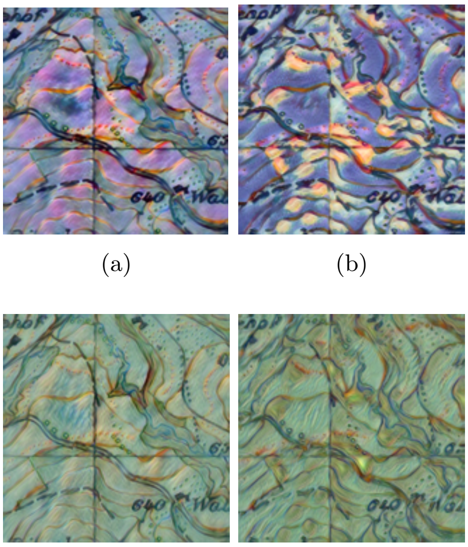
For the purpose of evaluating efficiency, we compared the training set with multiple styles of input to the training set with a single style of input, and we compared the training set with multiple content inputs to the training set with a single content input. Figure 4.5a and Figure 4.5e show the results of training with single content and single style input, which we refer to as specific models. Figure 4.5b and Figure 4.5f show the results of training with single content and multiple styles input. Figure 4.5c and Figure 4.5g show the results of training with multiple contents and single style input. Figure 4.5d and Figure 4.5h show the results of training with multiple contents and multiple styles input, which we refer to as general models.

Results from Figure 4.5 reveal significant differences between general and specific models.

4.1.4 Effect of Contrast Loss

The experiments discussed above all performed well in preserving style information, but they still have issues with the readability of the map. To enhance the contrast among different feature classes, different settings of contrast loss were compared.

The model tends to overfit and predict zero value when adding the contrast loss. Therefore, we



(a) (b)

(c) (d)

Figure 4.4: Effect of Different Style Loss, (a)(c) moment matching loss, (b)(d) gram loss.

experimented with different weights for contrast loss and discovered that the optimal ratio between contrast loss and content loss is 1:10.

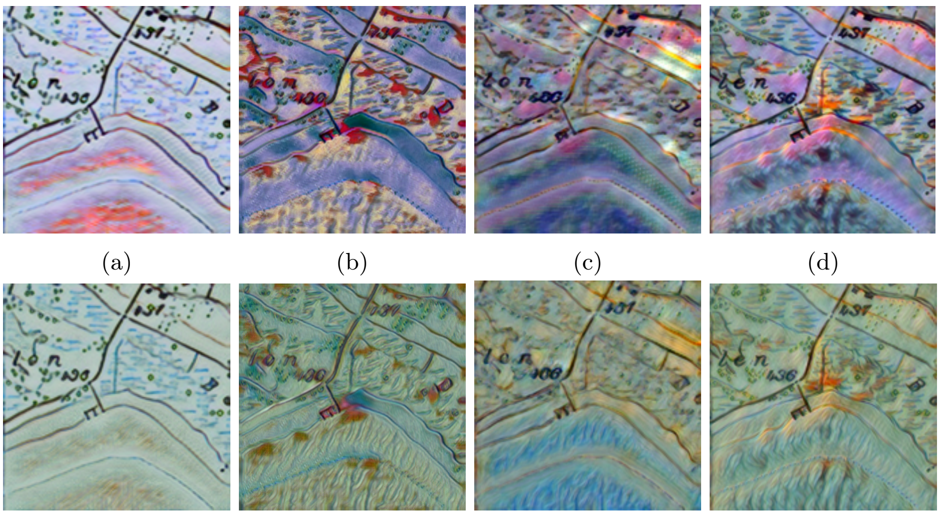
In Figure 4.6, we compared the models with and without contrast loss, and also compared the contrast loss setting using standard deviation versus cosine distance. Figure 4.8a, Figure 4.8d, and Figure 4.8g are without contrast loss. Figure 4.8b, Figure 4.8e, and Figure 4.8h are with contrast loss calculated from standard deviation. Figure 4.8c, Figure 4.8f, and Figure 4.8i are with contrast loss calculated from cosine distance. The output shows that with the inclusion of contrast loss, the color within a single feature class, such as the lake, becomes more uniform. In contrast, the color differences between different classes become more pronounced. For example, in comparison to Figure 4.6g, the wetland areas in Figure 4.6h appear bluer, while the lake areas appear pinker.

Figure 4.7 also reveals the effect of adding contrast loss. By incorporating contrast loss, the color contrast between the wetland and lake regions is improved.

4.2 User Study

In our user study, a total of 22 people participated. Participants were asked to fill in a questionnaire where they evaluated the results from different experiments and chose the results that best preserve the content image, style structure, and overall perform better. We also compared the performance between different styles. The result1 and result2 are from networks with AdaIN and AdaConv, both trained with a single style and multiple contents; result3 is from general model with moment matching style loss; result 4 is from general model with gram style loss; result 5 is from specific model.

According to the results of our user study for comparing different models, as shown in Figure 4.8 Figure 4.9 and Figure 4.10, both styles received similar responses from participants. The results obtained using specific models (i.e. those trained on single style and content inputs) were generally considered to have better performance in preserving content information. On the other hand, the results obtained using general models (i.e. those trained on multiple styles and contents of input) with gram loss were generally considered to have better performance in preserving style information and were overall considered better for the task of style transfer for maps.



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

(e) (f) (g) (h)

Figure 4.5: Effect of the different training sets, (a)(e) are trained with single content and single style input(specific models), (b)(f) are trained with single content and multiple style input, (c)(g) are trained with multiple contents and single style input, (d)(h) are trained with multiple contents and multiple style input.

In addition, we also asked users to evaluate the results of the different settings of contrast loss. The results shown in the figure indicate that when contrast loss was calculated using standard deviation, the contrast was most enhanced.

Moreover, users were presented with a stylized map and asked to identify all the map feature classes (i.e. river, building, etc.) they could. In the stylized map 4.12 we presented to them, everything appeared to be a shade of blue, leading them to have difficulty distinguishing road and river.

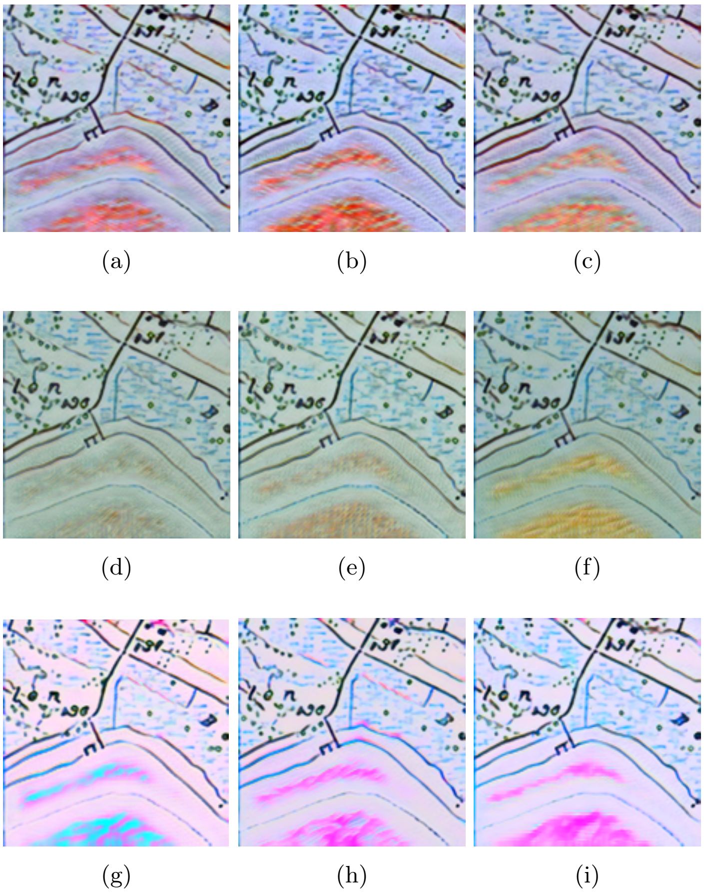


Figure 4.6: Effect of Contrast Loss, all trained with specific models, with gram loss as style loss,(a)(d)(g) are without contrast loss, (b)(e)(h) are with contrast loss calculated from standard deviation, (c)(f)(i) are with contrast loss calculated from cosine distance.

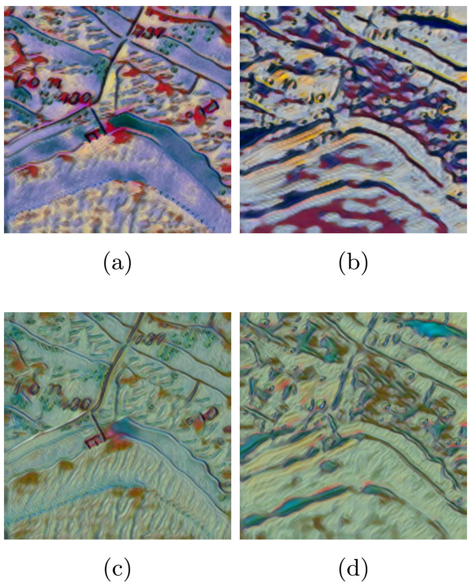


Figure 4.7: Effect of contrast loss, all trained with single content and multiple styles input, (a)(c) are without contrast loss, (b)(d) are with contrast loss calculated from standard deviation.

Figure 4.8: Result that preserves the content image better.

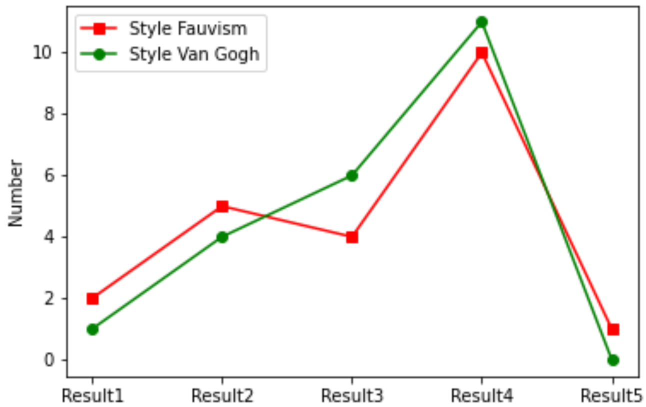


Figure 4.9: Result that preserves the style structure better.

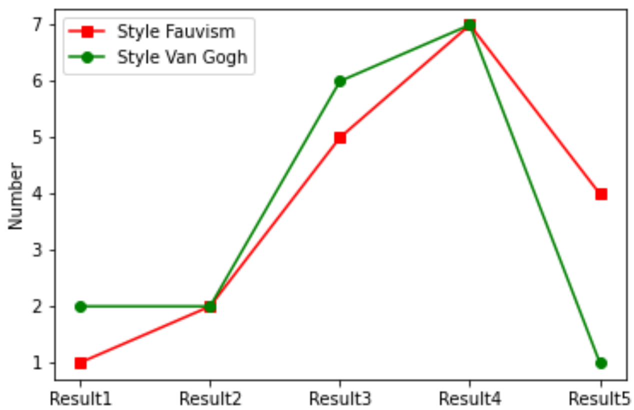


Figure 4.10: Result that has overall better performance in style transfer.

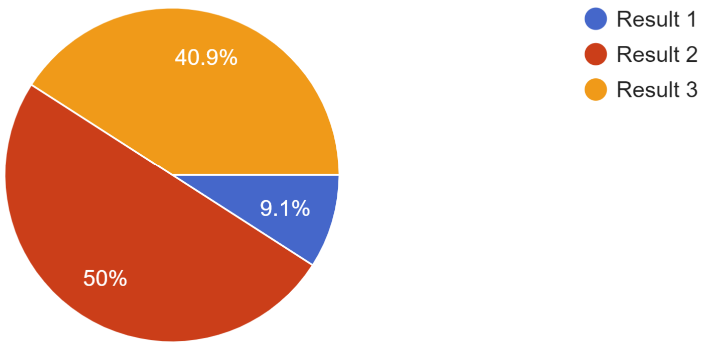


Figure 4.11: Result that evaluates contrast, "Result1" is without contrast loss, "Result2" is with contrast loss calculated using standard deviation, "Result3" is with contrast loss calculated using cosine distance.

Figure 4.12: Stylized map used to evaluate readability.

Chapter 5

Discussion

5.1 Network Structure

State of art methods, AdaIN and AdaConv, were implemented and compared. The results from AdaConv demonstrated superior performance in preserving the overall style structure of the input image, suggesting that AdaConv is able to capture both global and local structures from the style image, which is particularly useful in our task. By being able to generate contrast between different feature classes through the use of different textures, AdaConv is able to produce more visually appealing maps that are easier to interpret. This is a significant advantage over other style transfer methods that may struggle to effectively preserve the intricate details and nuances of the style image, leading to a loss of visual appeal and clarity in the resulting map.

5.2 Style Loss Function

In terms of the style loss function, our experiments showed that gram loss outperformed moment matching loss in our case. This is likely due to the fact that gram loss measures the style of an image by considering the correlations between different features, rather than just their individual values. By focusing on the relationships between features rather than their absolute values, gram loss is able to capture higher-level correlations that may be lost when using other style loss functions. This results in a more accurate representation of the style of the input image, leading to more visually appealing and faithful style transfer results.

5.3 Contrast Loss Function

While our contrast loss was able to slightly enhance the contrast between different feature classes, it was unable to change the texture of the map. This suggests that there may be potential for improving contrast and readability by altering the texture of the map rather than simply adjusting the contrast.

5.4 Training set

We also compared the performance of models trained on different types of input and found that general models, which were trained on multiple styles and contents, tended to perform better in preserving the style structure of the input image and were considered to have overall better performance. In contrast, specific models, which were trained on a single style and content, tended to perform better in preserving content information. This may be because when training specific models, the network becomes highly specialized and fine-tuned to the specific content and style of the input image, leading to an improvement in content preservation. However, this specialization may come at the expense of the model's ability to handle a wider range of styles and content. On the other hand, when training general models, the model may be more flexible and able to adapt to a variety of styles and content, leading to better preservation of the overall style structure.

5.5 User Study

In our user study, we asked participants to evaluate stylized maps and found that they had strong intuitive associations with certain colors. For example, they tended to identify blue as representing watercourses and green as representing vegetation. This indicates the importance of considering the empirical understanding of color in cartography.

An interesting observation from our user study is that while readability is generally considered the most important aspect of maps, a significant number of people still prefer style transfer results that preserve more of the original style texture at the expense of readability.

Chapter 6

Conclusion

6.1 Summary

In this project, we conducted several experiments to apply neural style transfer for maps.

Based on the findings discussed in this report, it appears that neural style transfer can be applied successfully to stylize topological maps. Among the methods tested, AdaConv demonstrated the best performance in preserving the style structure of the input image, while also being able to capture both global and local structures. Gram loss was found to be the most effective style loss function, outperforming moment matching loss.

Additionally, it was observed that general models, which were trained on a diverse range of styles and contents, tended to perform better in preserving the overall style structure of the input image. Specific models, on the other hand, were more effective at preserving content information.

Overall, these results suggest that it is possible to use neural style transfer to enhance the visual appeal and clarity of topological maps, and that there is potential for further improvement by fine-tuning the model architecture and training process.

Our work represents a thorough investigation of style transfer for maps. Previous research in this area [13, 18, 4, 2] has largely focused on transferring style from satellite images to maps, but our study specifically addresses the problem of stylizing already existing topological maps. This is a novel and important direction for research, as it has the potential to enhance the visual appeal and usability of topological maps in various applications.

6.2 Outlook

There are several avenues for future research that could build upon and expand upon our current pipeline:

- First, one possibility is to further refine the model architecture and training process to improve the performance of style transfer for maps. This could involve exploring alternative encoder and decoder networks, or experimenting with different training strategies such as data augmentation or transfer learning. We currently only use a pre-trained VGG-19 network as our encoder. This could be changed to a different encoder network that is more suited to our task and better able to extract map features.
- Second, style features could be extracted based on different colors and distinct kernels could be predicted for different feature classes. By applying these kernels in the appropriate layers

of the decoder network, it may be possible to generate maps with enhanced contrast between different features, improving the overall visual appeal and readability of the map. This approach could be particularly useful in cases where the original map has low contrast or indistinct features, and could potentially be applied to a wide range of map types and scales.

- Third, our current model has only been tested on small regions of maps. In the future, it would be beneficial to apply it on maps at multiple scales. This would require us to improve the robustness of our approach.
- In addition, it may be worthwhile to explore the use of style transfer for maps in a range of different applications, such as personalized map design or visualization of geospatial data. This could involve conducting user studies or experiments to evaluate the effectiveness of style transfer in these contexts and identify any potential benefits or drawbacks.

Overall, the development of effective and efficient methods for style transfer for maps has the potential to enhance the visual appeal and usability of topological maps in various applications, and there is considerable room for further research in this area.

Bibliography

[1] P. Chandran, G. Zoss, P. Gotardo, M. Gross, and D. Bradley. Adaptive convolutions for structure-aware style transfer. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7968–7977, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[2] Xu Chen, Songqiang Chen, Tian Xu, Bangguo Yin, Jian Peng, Xiaoming Mei, and Haifeng Li. Smapgan: Generative adversarial network-based semisupervised styled map tile generation method. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4388–4406, 2021.

[3] Sidonie Christophe and Charlotte Hoarau. Expressive map design based on pop art: Revisit of semiology of graphics? *Journal of the Brazilian Computer Society*, pages 61–74, 2012.

[4] Sidonie Christophe, Samuel Mermet, Morgan Laurent, and Guillaume Touya. Neural map style transfer exploration with gans. *International Journal of Cartography*, 8(1):18–36, 2022.

[5] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style, 2016.

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018.

[8] Magnus Heitzler and Lorenz Hurni. Cartographic reconstruction of building footprints from historical maps: A study on the swiss siegfried map. *Transactions in GIS*, 24(2):442–461, 2020.

[9] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. pages 1510–1519, 10 2017.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

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

[12] Yuhao Kang, Song Gao, and Robert E. Roth. Transferring multiscale map styles using generative adversarial networks. *International Journal of Cartography*, 5(2-3):115–141, 2019.

[13] Jingtao Li, Zhanlong Chen, Xiaozhen Zhao, and Lijia Shao. Mapgan: An intelligent generation model for network tile maps. *Sensors*, 20(11), 2020.

[14] Hubert Lin, Mitchell van Zuijlen, Sylvia C. Pont, Maarten W. A. Wijntjes, and Kavita Bala. What can style transfer and paintings do for model robustness?, 2020.

[15] Yilin Tao. Image style transfer based on vgg neural network model. In *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pages 1475–1482, 2022.

[16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images, 2016.

[17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, 2017.

[18] Mingguang Wu, Yanjie Sun, and Yaqian Li. Adaptive transfer of color from images to maps and visualizations. *Cartography and Geographic Information Science*, 49(4):289–312, 2022.

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

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

If Van Gogh or Picasso Painted Maps - Explorations of Neural Style Transfer Applied to Maps

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
Wang	Rushan

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	Signature(s)
Zurich, 19, 01, 2023	Rushan Wang

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.