UNSUPERVISED STRUCTURE-CONSISTENT IMAGE-TO-IMAGE TRANSLATION

Shima Shahfar

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Science in Computer Science
Concordia University
Montréal, Québec, Canada

April 2022

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Shima Shahfar**

Entitled: **Unsupervised Structure-Consistent Image-to-Image Translation**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Yiming Xiao ______________ Chair
Dr. Adam Krzyzak ______________ Examiner
Dr. Yiming Xiao ______________ Examiner
Dr. Charalambos Poullis ______________ Supervisor

Approved By Dr. Leila Kosseim, ______________
Chair of Department or Graduate Program Director

__________ 2022 ______ ______________

Dr. Mourad Debbabi, Interim Dean
Faculty of Engineering and Computer Science

Abstract

Unsupervised Structure-Consistent Image-to-Image Translation

Shima Shahfar

There have been significant advances in designing deep networks for complex computer vision tasks. One that is of considerable importance is image understanding through pixel-wise classification, i.e. semantic segmentation. Despite the advances, self-supervised algorithms have many limitations and challenges, with perhaps the most significant being generalization. This thesis introduces a method based on generative models as a practical approach for addressing these shortcomings. First, we analyze several semantic segmentation methods to gain insight into their limitations. We investigate the effectiveness of one of the state-of-the-art methods on two different problem settings. The latter part of the thesis introduces an alternative approach using generative adversarial networks and autoencoders for image-to-image translation. The main idea is encoding an image into two latent codes to represent structure and style. We propose a new approach to enforce structure-consistency without requiring semantic labels to disentangle the two latent codes. We further show how this would result in a more detailed style transfer and image manipulation. Finally, we present results on multiple datasets and discuss how our approach can be practical in real-world applications. Our experiments demonstrate that our approach performs better than the baselines -or, in the worst-case, gives comparable results- while solving some of the shortcomings in tasks requiring a semantic mask.

Acknowledgments

I would like to thank my thesis advisor Charalambos Poullis for his mentorship and many helpful discussion, for giving me the chance to work on interesting projects. I would like to extend my gratitude to my mother and my siblings for their love and support. I would not have gotten this far if wasn't for them. Finally, the work reported in this thesis would not have been possible without the financial support from: DND/NSERC

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem of image understanding has been an important and challenging task in computer vision. It can have a considerably important role in different real-world scenarios, from object detection and image classification to semantic segmentation and image recognition. This thesis focuses on semantic segmentation, which classifies input images at pixel-level and can provide fine-grained predictions. Recent advances in deep learning raise new possibilities to achieve better scene understanding through semantic segmentation. However, the expense of creating datasets can be very high in any domain. On the other hand, a vast amount of unlabeled data is available and can be used for training without the further overhead. That is why trying to find a cost-effective alternative can have a significant impact in practical settings. Thus, one of this work's objectives is to use the available data to generate synthetic datasets for the supervised semantic segmentation task. Techniques for supervised semantic segmentation tasks usually struggle to generalize when tested on a different domain regardless of how visually close the images are, caused by a lack of a broad range of training data or considerable bias in class distributions.

In this thesis, we study how recent advances in generative adversarial networks can be applied to resolve training data issues. Chapter 2 introduces the concepts necessary to understand the work. Chapter 3 contains our studies on semantic segmentation methods and their limitations. Chapter 4 introduces an unsupervised technique for image-to-image translation that can generate high-quality structure-consistent images by learning better feature embedding. We further show our results and analysis on image generation, style transfer, and image reconstruction quality. We also provide a quantitative comparison with state-of-the-art methods and explain how this idea can be used in different applications. Finally, in Chapter 5, we discuss the conclusions of our works and propose future works.

Contributions In this work, we address the following research questions:

- How can we improve the generalization capability of supervised semantic segmentation models and improve their performance? Researchers have found different approaches to improve generalization in supervised tasks. Some of the common approaches are designing methods for better generalization and taking biases in the training data into consideration while designing the architecture, creating larger datasets for training that can broaden the data distribution and improve generalization, and generating synthetic data. The first approach can reduce the effect of bias, but it cannot resolve the generalization issue since the data distribution is still limited to the training set. The second approach is very costly and can be impossible to achieve in many domains, while the third approach puts most of the pressure on the algorithm design process and gives researchers the option to choose what data characteristics matter to them the most for their specific application. We focus on generating synthetic images for most of this thesis.

- Can we preserve the structure and spatial information without direct supervision in image-to-image translation tasks? Recent image-to-image translation methods [39,40,45] have shown impressive results on single object images. That being said, to preserve the structure, many works [48, 65, 100] follow semantic image synthesis problem setting to generate an image conditioned on a semantic mask, but we do not want to limit our method by requiring pixel annotations. The idea is that instead of conditioning the image generation on the semantic mask, we encode the input image into semantic and style information and learn the embedding. We then use a decoder to render the image based on the extracted embedding.

- Can a better understanding of the geometry of the objects result in transferring class-based attributes while synthesizing an image? We find that having a better understanding of the geometry to be beneficial in most circumstances, specially for image manipulation. This would allow local manipulation of the objects. It also act as an auxiliary information that leads to more accurate attribute transfer. To ensure disentangling structure from texture we propose a new module to suppress any texture-related features in structure code. We achieve this through gradient reversal layer [16].

- Would this method perform well in complex domains? To answer this question we evaluate our proposed method on aerial imagery which is known to be a complex domain. Our results show that our method is able to generate structure-consistent images and transfer texture very well.

Notes on Contributions I am the first author of the papers presented in this thesis. My role in this work was to form the ideas, design and implement the algorithms and pipelines, conduct the experiments and studies, and the writing of scientific publications. The contributions of co-authors include manuscript review, supervision, analysing studies, and technical support.

Chapter 2

Background

Abstract

In this chapter, we present different strategies for learning an image representation. Before presenting any learning method, we introduce the basics of feed-forward neural networks. The chapter continues with the discussion on the use case of these models for image understanding and a general technique to train them. We also discuss two different supervised approaches in learning representations, namely image classification and regression, followed by a short introduction of different approaches in learning representations. Then, we present an overview of object detection, semantic segmentation, generative models, and generative adversarial networks. We close the chapter by discussing transfer learning, domain adaptation and image to image translation.

2.1 Neural Networks

The inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen, defines artificial neural networks as ”a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.” There is a myth that neural networks have been inspired by the idea of replicating the biological neural systems but after improvements in computational hardware systems and mainly the work of Alex Krizhevsky et al. in AlexNet [46] inspired many scientists to achieve better results in machine learning tasks.

2.1.1 Artificial Neuron

Classical artificial neurons motivated by biological neurons, which are the basic computational units of the human nervous system, are connected by approximately 10^{15} synapses. Each neuron works as a function that receives single or multiple inputs and produces an output.

To drive the mathematical form of a artificial neuron, consider a set of features $x = \{x_1, x_2, ..., x_i\}$ and a set of weights corresponding to each feature $w = \{w_1, w_2, ..., w_i\}$ with a scalar value as bias called b, an artificial neuron $f(x)$ can be defined in vector notation as:

$$f(x) = h(w^T x + b)$$

where $h(.)$ is an activation function or in biological neurons the firing rate.

Activation Functions

Activation functions are usually continuous functions applied element-wise on a pre-activation. Considering the function defined above, an activation function $h(.)$ determines whether the neuron should be activated or not by applying an activation function (or non-linearity) on its pre-activation values. We call $w^T x + b$ a pre-activation which here is simply a linear weighted sum. Most of the activation functions are simple functions; however, given that the network has enough capacity based on universal approximation theorem [64], they can make the network approximate any complex function. Some of the commonly used activation functions are introduced below.

Sigmoid

The Sigmoid function [24] maps any pre-activation values to the range of $[0, 1]$, and it can be represented by the following formula, which creates an ”S”-shaped curve.

$$sigmoid(z) = \frac{1}{1 + \exp^{-z}}$$

where z denotes a pre-activation.

Hyperbolic Tangent (Tanh)

The Tanh function [21] squashes any pre-activation values to the range of $[-1, 1]$. It defines by the following formula

$$Tanh(z) = \frac{\exp^z - \exp^{-z}}{\exp^z + \exp^{-z}}$$

Rectifier Linear Unit (ReLU)

This function returns zero for all negative inputs, and for positive inputs, it will return the same value as its pre-activation. ReLU function [63] can be formulated as $\max(0, z)$ and it became more popular during the past few years.

Leaky-ReLU

The idea behind leaky-ReLU [60] is to keep the weights for negative values during backpropagation and so they added a small negative slope to ReLU.

$$LeakyReLU(z, \lambda) = \max(0, \lambda z) \quad , \lambda > 0$$

Technically ReLU is a particular case of leaky-ReLU where $\lambda = 1$.

Exponential Linear Units (ELU)

This activation function was introduced in [11], and it is another variation of ReLU which tries to decrease bias shifts during the training process by pushing mean activation towards zero.

$$ELU(z, \lambda) = \max(0, z) + \min(0, \lambda(\exp(z) - 1))$$

Softmax

Softmax function is used to calculate the probability distribution over a vector of outputs and then returns the probabilities. It is one of the most common activation functions for multi-class classification tasks. The softmax formula is as follows:

$$f(z_i) = \frac{\exp(z_i)}{\Sigma_j \exp(z_j)}$$

where all the z_i values are the elements of the input vector. The sum of returned probabilities of Softmax is always equal to 1.

2.1.2 Convolutional Neural Networks

Convolutional Neural Networks or ConvNets are very similar to standard neural networks that have been around for a long time. The only difference is that in ConvNets were designed for tasks on images, so they can encode some of the properties of this domain to the architecture design by using kernels. Using different kernels with different weights allows the network to extract different types of features from input, which can then be used for a more efficient and informed forward function. Also, ConvNets arranges its neurons in three dimensions which reduces the number of parameters by a significant factor.

2.1.3 Loss Functions

A loss function, also known as a cost function in mathematical optimization literature, is a function that maps one or more variables to a single real number [85]. In mathematical optimization, the loss function is a function that we want to minimize. It is also true about machine learning and training the neural network, but machine learning (ML) is different from optimization because, in ML, we do not care about the objective function. What we care about in machine learning is the test result. However, in theory, we use the same problem formulation as optimization problems to train our neural networks.

Empirical Risk Minimization [82] is the process of training a neural network by minimizing the expectation of the loss (or maximizing its negative). The objective here is to minimize the loss between the true output and the predictions of the network.

Empirical Risk Minimization is defined as follows:

$$L(x, \theta) = \arg\min_{\theta} \frac{1}{N} \Sigma_n l(f(x^{(n)}; \theta), y)$$

where x is input, y is the target, θ refers to the parameters of the model (usually weights and biases or offset) and l can be any loss function defined for the task at hand. Assuming that we have N paired data point for training $(x^{(n)}, y^{(n)})$ in supervised setting, $x^{(n)}$ refers to the n-th input data. If we add the regularization term to Empirical Risk function defined above, we can create a structural risk minimization function as follows:

$$L(x, \theta) = \arg\min_{\theta} \frac{1}{N} \Sigma_n l(f(x^{(n)}; \theta), y) + \lambda \Gamma(\theta)$$

where Γ is a regularizer which is penalizing certain values of θ. Some popular regularizers are L1, L2, and dropout [76].

2.1.4 Training

Usually, when dealing with a machine learning task, we do not know the function that transfers input x to the output y, and because of that, we cannot calculate the optimal solution directly using closed-form solutions. Hence, there is no guarantee that a neural network will find the global optimum solution as it can be non-convex. In practice, it is common to use gradient-based optimization methods such as gradient descent and second-order derivatives, to name a few. We will cover some of the most useful algorithms for training a neural network in the following section.

2.1.5 Optimization Algorithms

Our goal in the optimization step is to find W that minimizes the loss function and let us qualify the network parameters in each step. We will now motivate and slowly develop an approach to optimizing the loss function. We will explain some of the popular optimization algorithms for deep neural networks.

Gradient Decent

Gradient Decent is an optimization algorithm where we take a step in the opposite direction of the gradients. It is known as the most common way of optimizing neural network loss functions. We can formulate the update step i as follows:

$$\theta_{i+1} \leftarrow \theta_i - \eta \nabla_{\theta_i} L(x, \theta_i)$$

in which ∇ is the learning rate. Although calculating gradients at each step and moving towards the minimum is an interesting idea, gradient descent needs to sum the gradients over the entire training set to take a single step, which is usually not suitable in practice. A standard method to prevent this is to take smaller batches from the data and do the gradient update step for these smaller batches, usually containing less than 256 examples depending on the task and accessible computational power.

Stochastic Gradient Decent (SGD)

One possible solution for computation complexity of gradient descent is to perform a stochastic gradient descent, where we iteratively follow the direction of the slope starting from a random point until we find a minimum using one sample each time. Vanilla SGD is still probably one of the most popular methods of training deep learning. SGD randomly picks one data point from the training set at each step and updates the gradients based on this sample. A common mistake is to call the method they are using SGD when they are sampling more than one point at each step. Based on the literature, if we use a small number of samples, we use mini-batch gradient descent or MGD.

SGD with Momentum

One iteration in stochastic gradient descent with momentum consists of the following steps, with the difference that it now uses a velocity term based on the network's previous update [69]:

1. Calculate gradient estimate:
$$h \leftarrow \frac{1}{m} \nabla_{theta} \sigma_i L(f(x^{(i)};\theta), y(i))$$
2. Calculate velocity update:
$$v \leftarrow \alpha v - \epsilon h$$
3. Apply update:
$$\theta \leftarrow \theta + v$$

SGD with Nesterov Momentum

Nesterov momentum [77] is a modified version of the momentum where the velocity term will be applied first and the gradient calculation will be applied as a next step. It can be formulated as follows:

1. Calculate velocity update:
$$\hat{\theta} \leftarrow \theta + \alpha v$$

2. Calculate gradient estimate:

$$h \leftarrow \frac{1}{m} \nabla_{theta} \sigma_i L(f(x^{(i)}; \hat{\theta}), y(i))$$

3. Calculate velocity update:

$$v \leftarrow \alpha v - h$$

4. Apply update:

$$\theta \leftarrow \theta + v$$

v is the velocity term.

Adam

Adam [43] is an adaptive optimization algorithm in the sense that it uses the estimation of second-order derivative and adapts to different learning rates for different parameters of the model. In order to take advantage of momentum, Adam keeps a moving average of the gradient.

2.2 Representation Learning

The data representation of the features we are using to train a machine learning algorithm is an essential characteristic of the data that needs to be considered when designing a new algorithm. Not long ago, if one wanted to have a well-designed machine learning algorithm to perform well on the data, the best way to ask a human expert who used their prior knowledge to identify the most important features in that data and manually engineer its features for the data. However, it is usually not that easy for machine learning engineers to have access to experts in all different fields and create the features in this way. It is also costly and time-consuming. The goal of representation learning [3] is that we would like our algorithms to automatically learn the best features for the specific tasks and data.

2.2.1 Supervised Learning

In supervised learning, we have access to labels or annotations during training, depending on a task. Having annotations let us better approximate and create the mapping function from inputs to outputs. It also sometimes helps to define more meaningful loss functions.

Classification

Classification is one of the widely used supervised algorithms. In classification tasks, we ask computers to predict the likelihood of which class the input data belongs. A classic example to introduce classification is the task of spam detection. Any incoming email can be classified as spam or not spam which means we only have two classes, and so the task is a binary classification. A classification algorithm is usually asked to create a function $f : IR^n \to \{1, 2, ..., k\}$ from input x to the corresponding class number. There are some variations of the classification algorithm where f outputs a probability distribution over classes, and then we need to apply a post-processing step to find the class id. Cross entropy is a common choice of loss function when we have a classification task.

Regression

Regression is a supervised learning task where we expect the model to predict continuous outputs $f : IR^n \to IR$. The loss or risk functions in regression tasks are usually basic functions like least-square or L2-norm. You may also see higher norms being used when facing a regression task. We can formulate a general loss function for regression tasks as follows:

$$L(f) = \|target - f(x)\|_p^p$$

2.2.2 Unsupervised Learning

Unsupervised learning is when no annotations are available for training the network. This problem setting becomes very interesting and extremely helpful in domains where more data is available, but we do not have access to annotations or are too expensive. This technique allows the network to learn at its own pace and learn common patterns from unlabeled data. The main assumption is that images are very rich in information, however, many loss functions designed for supervised learning provide a sparse feedback to the network. Unsupervised learning aim to discover common patterns and features solely based on training set.

2.2.3 Self-supervised Learning

Achieving good performance in supervised tasks usually depends on the amount of labelled data available for the task at hand. However, collecting and creating manual labels are expensive and time-consuming. On the other hand, considering the data being produced in our daily lives worldwide, it would give us a considerable boost in performance if we can find a way to use these unlabeled data.

The self-supervised learning method is designed to allow the network to learn from unlabeled data in a somewhat special setting in which they use an unsupervised method to get the labels for part of the data and then use a supervised learning task to predict only a subset of information using the rest of the data available.

2.3 Object Detection

Object detection is designed to recognize different objects in an input image, predict the class they belong to, and find a bounding box around each object that appears in the image. It is also one of the common computer vision tasks that has been around long before deep learning and convolutional neural networks. Before the CNN era, people attempted to solve the problems in multiple steps. Starting from an image, they have first extracted the edges and then using a feature extractor algorithm such as SIFT or Histogram of oriented gradients, they would have created a pyramid to localize different objects.

Nowadays, we have two different types of object detection algorithms: two-step algorithms and one-step algorithms. Two-step algorithms first predict the bounding box around the objects, and then in the second step, they predict the class associated with the object inside the bounding box using a classification algorithm. In the first step, these methods usually start with a Region Proposal Network (RPN) to extract many regions most likely to contain the object. These algorithms perform better than one-step methods, but they are slower in comparison because of the bottleneck in RPN step. RCNNs [19], Fast RCNNs [18] and Faster RCNNs [68] are few examples of two-steps algorithms. On the other hand, one-step algorithms are designed for real-time applications and so they are trying to merge the two steps of detection and recognition to one. YOLO [67], SSD [57], and RetinaNet [54] are some examples of one-step object detection algorithms.

2.4 Semantic Segmentation

Semantic segmentation is a pixel-wise classification in which we label each pixel of an image with a corresponding class of the object represented in the image. In semantic segmentation, we do not separate instances of the same object from each other, and we behave them all the same. It is worth mentioning that there is a different problem setting that distinguishes between different instances of the same object, known as instance segmentation.

The task in semantic segmentation can be defined as follows. Having an image, either a RGB ($height \times width \times 3$) or a grayscale ($height \times width \times 1$), our goal is to output a segmentation map ($height \times width \times 1$) where each pixel contains a class label represented as an integer value corresponding to the class it is representing.

2.5 Generative Models

In this section, we introduce generative models related to our work and their formal definitions.

2.5.1 Generative Adversarial Networks

Generative Adversarial Networks by [22] is a framework for estimating data distribution and generating samples fortunately near to original distribution. The model consists of two parts, a generator G which generates new samples and tries to fool the discriminator D. The idea is that G is generating samples that are close to the original data distribution we showed to the discriminator, and in the training process, it will update itself to generate better samples using the discriminator output.
Ian J. Goodfellow refers to this framework as a minimax two-player game in which G try to minimize the probability of D recognize the fake samples and D try to maximize the probability of assigning the correct label. The objective function can be written as follows: $\min_G \max_D V(D, G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$
If the model has enough capacity, after training D and G simultaneously, we will reach a point that the discriminator cannot differentiate between the two distributions.

Conditional GANs

Generative adversarial networks can be extended to conditional generative models [62] by feeding additional information c into either the discriminator or generator. c can be any information such as edge mask for semantic segmentation task or class labels for classification. By doing so, the generator can use prior noise $p_z(z)$ and additional information c to create a hidden representation, and the discriminator will be able to use the information provided as an input for better discrimination. The objective function can be represented in what follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x|c)] + E_{z\sim p_z(z)}[\log(1-D(G(z|c)))]$$

2.5.2 VAEs

Like autoencoders, variational autoencoders consist of an encoder and a decoder. However, in VAEs, we encode the input to a distribution, whereas, in autoencoders, we encode the input into a latent code. After encoding an image, we sample a point from the distribution and pass it to the decoder.

2.6 Image-to-Image Translation

The image-to-image translation task aims to learn a mapping to transfer an image from one domain to another while preserving general content. There has been much progress in image-to-image translation problems. Mostly due to the wide range of applications it can have in computer vision like style transfer, domain adaptation, image manipulation, segmentation and pose estimation to name a few.

One can say this task has been popularized by the work of [35] where they proposed a method based on Conditional GANs [62]. The novelty of their work is that they learn the mapping from one domain to another using paired images and learn a loss function in the training process. This additional feature makes the architecture suitable for a wide range of problems where designing a loss function is difficult.

2.7 Conclusion

In this chapter, the fundamentals of neural networks, generative models and famous computer vision tasks have been covered. We also discussed different optimization methods and common problem settings.

Chapter 3

Semantic Segmentation

Abstract

This chapter discusses the effectiveness of supervised semantic segmentation for geospatial data and explores its limitations. We build upon Gated Shape CNNs for Semantic Segmentation (Gated-SCNN) [78], and we reduce the training time of this method by optimizing the forward path. We show our results on two datasets, namely, Inria and Spacenet6.

3.1 Introduction

The idea behind GSCNN is that all types of features cannot be processed through one convolutional neural network. Towards this, they proposed a different branch to extract shapes and boundaries. Predicting pixels around boundaries is usually where many segmentation networks fail. This new branch can be added to any existing network to achieve better results since the information learned in the shape branch will be concatenated with the traditional segmentation branch. We chose this method due to the superiority and high performance they achieved. We present fast-GSCNN by proposing some changes to the code and moving the shape branch processing on GPU instead of CPU. We improved the technique's performance by revising the activation function of the final layer, which resulted in faster convergence. First, we evaluate the architecture on Inria [61] and SpaceNet6 [72] datasets. We then discuss our findings, observations and future works.

3.2 Methodology

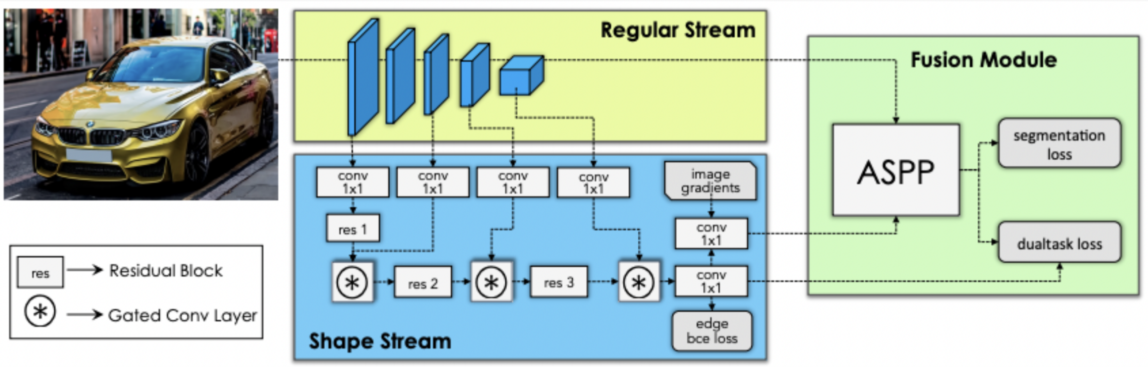

Figure 1: A representation of GSCNN [78] architecture. The photo is borrowed from the original paper.

GSCNN [78] is an architecture with two-stream designed to improve the quality of segmentation models by better predicting masks around object boundaries. The regular stream can be any feedforward convolutional network. We use ResNet-101 architecture [26]. The shape stream would first extract edges by calculating the image gradient. The architecture of this stream consists of 1 × 1 convolutions, a

City	IoU	Accuracy	SOTA IoU	SOTA Accuracy
Bellingham	67.56	96.58	74.63	97.47
Bloomington	67.71	96.89	80.80	98.18
Innsbruck	72.06	96.64	79.50	97.58
San Francisco	76.14	91.96	81.85	94.08
Eastern Tvrol	73.47	97.58	81.71	98.39
overall	72.87	95.93	80.32	97.14

Table 1: Quantitative comparison of our results on Inria datasets with the state of the art [5].

few residual blocks and gated convolution layers to combine different components. They improved Deeplab V3 [6] by 1.5 in terms of mIoU and 4 percent in terms of F-boundary score. The training is an end-to-end process performed by the fusion module via combining the two branches and learning how to predict both the segmentation mask and boundary mask simultaneously. The Dual-Task Regularizer helps to exploit the boundary space to make better predictions. The Fusion module takes the output of the two branches as an input and combines the two using Atrous Spatial Pyramid Pooling [25] and produces a categorical distribution of different classes.

Objective function We jointly train the two streams using the Fusion module. $L = \lambda_{main} L_{CE}(s, s_{pred}) + \lambda_{shape} L_{BCE}(b, \hat{b}) + \lambda_{reg} L_{reg}$ where s is the ground truth segmentation map, and b is the grounth truth boundary map. We calculate cross entropy between the ground truth and predicted labels.

3.3 Experiments

Datasets.

- Inria [61] is an aerial imagery dataset for building segmentation. The training set contains 180 images with 5000×5000 resolution from 5 cities. Each image covers an area of approximately $1500m \times 1500m$. The test set contains 180 images of the same size collected from 5 cities not part of the training set.

- SpaceNet6 [72] is a multi-sensor dataset that consists of two modalities that

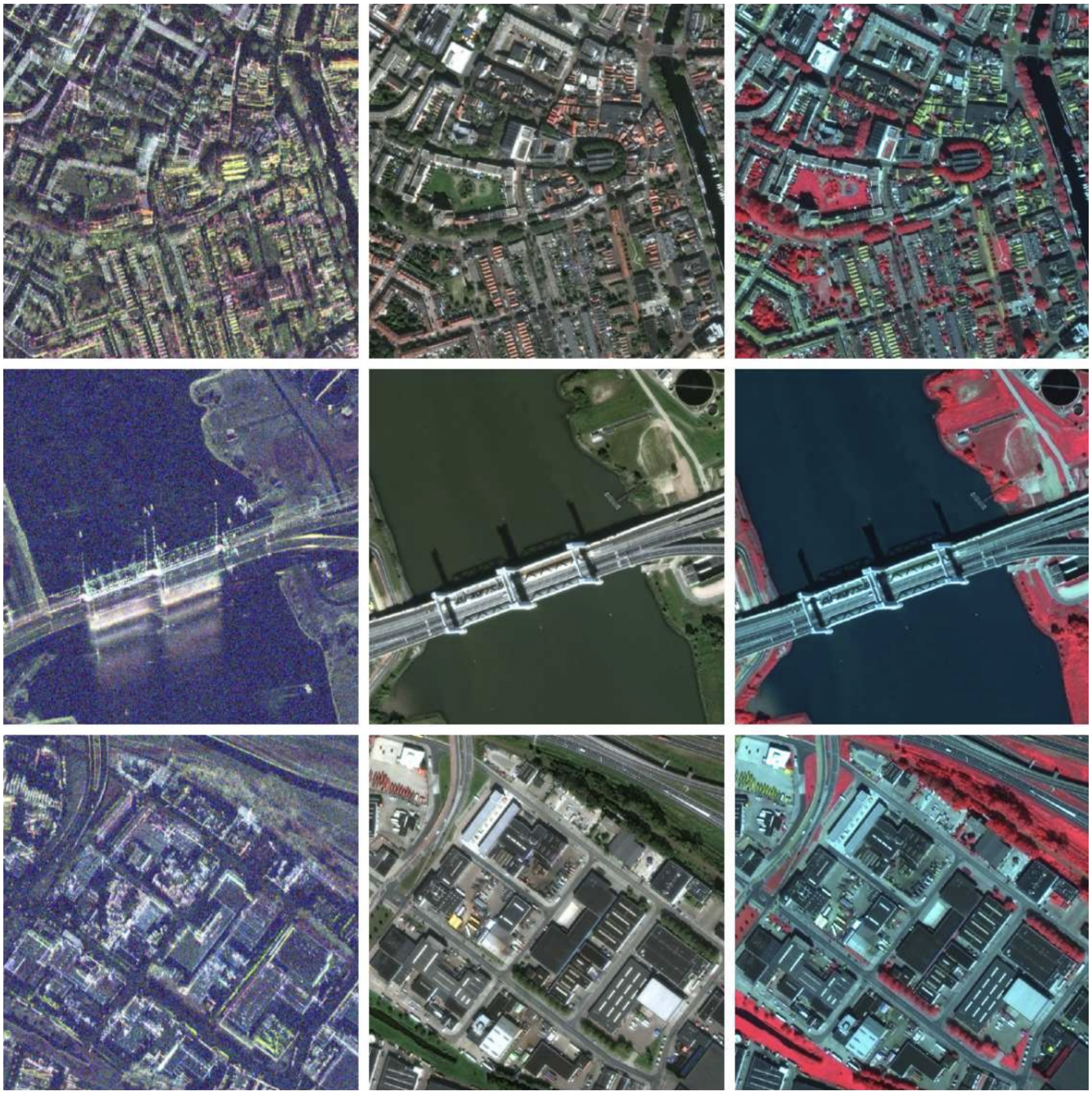

Figure 2: Data visualization from [72]. "Left: SAR Intensity (HH, VV, VH). Center: Visible Spectrum Imagery (R,G,B). Right: False Color Composite Imagery (NIR, R, G)."

can be used during training. One is collected by Synthetic Aperture Radar (SAR) sensors with four polarizations (HH, VV, HV, VH), and the other is RGB imagery collected using Maxar Worldview-2 satellite. The annotations for building footprints are provided by the 3DBAG dataset [**?**, **?**]. The challenge of this dataset is that, due to the high costs of data collection, RGB imagery is only available at training time. Figure 2 shows some samples of the dataset.

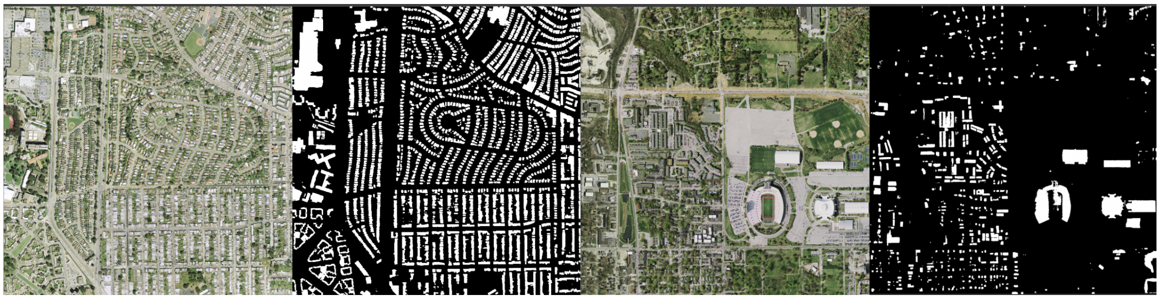

Figure 3: Qualitative results of our fast-GSCNN on Inria test set. The images shown are from San Fransisco and Bloomington.

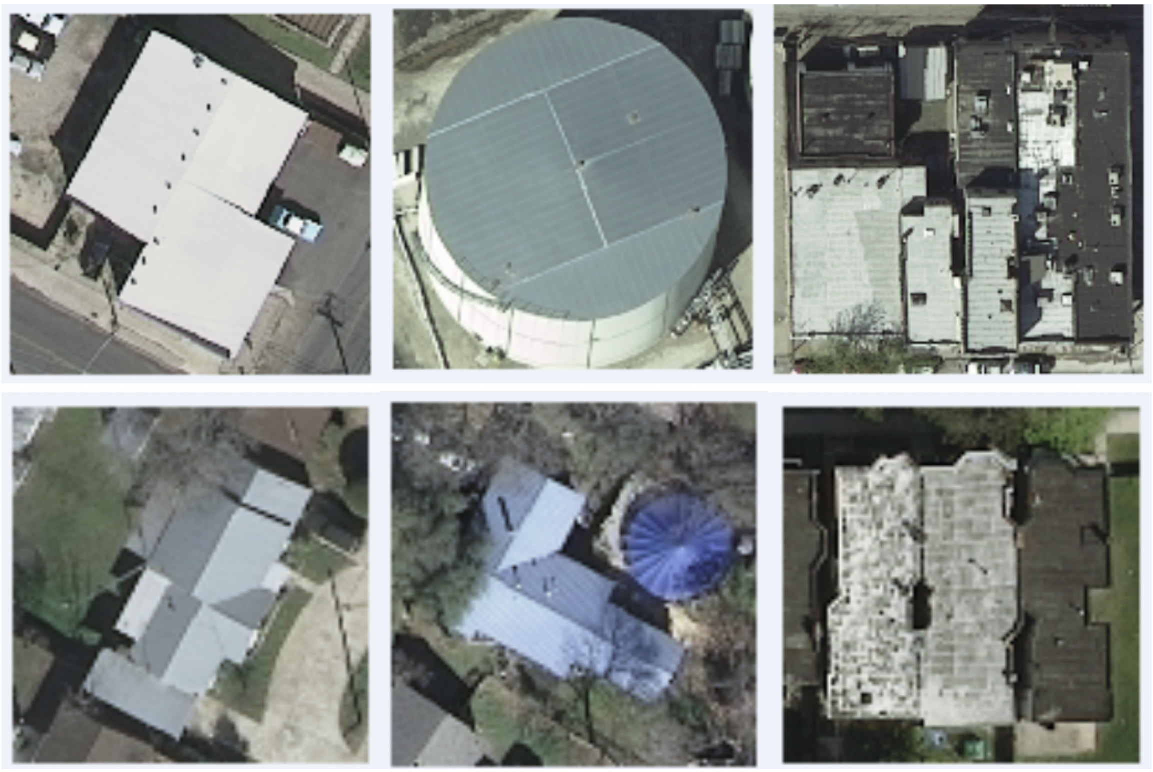

Figure 4: Example roof tops extracted from Inria. The variety of roof tops in Inria is limited since it only contains images from 5 cities. These variations are one of the reasons we believe that a network trained on one city cannot be generalized to samples from other cities not seen during training.

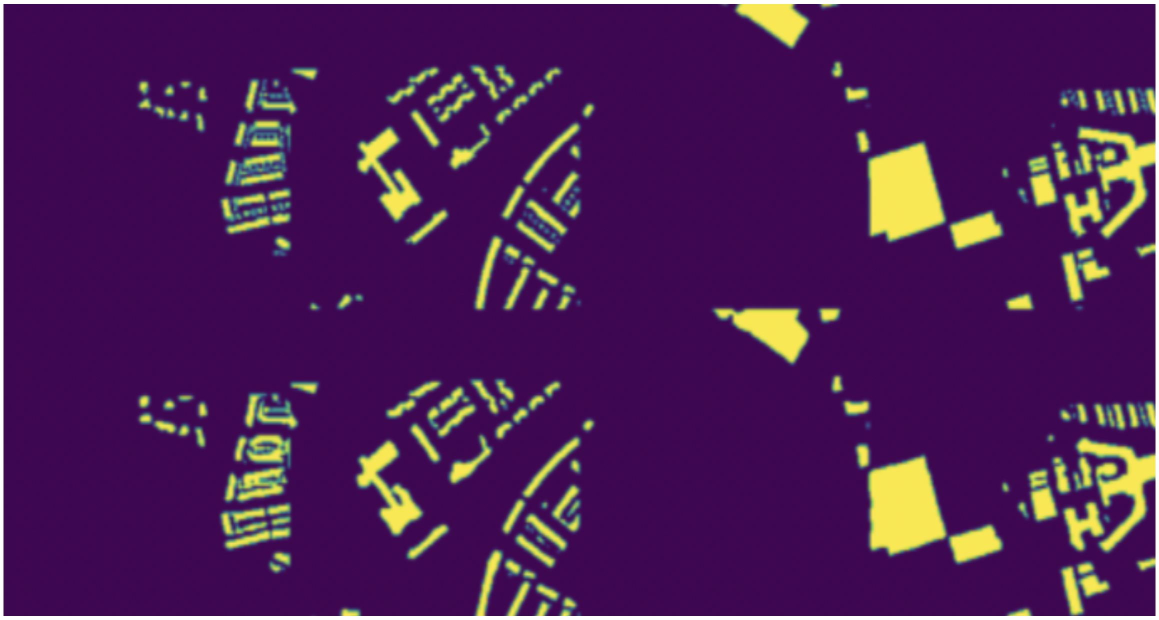

Figure 5: This figure provide an example of how fast-GSCNN perform on SpaceNet6 dataset. First row shows ground truth and second row represents our predictions.

Metrics	Results on validation set
Mean IoU	0.8375
Precision	0.8903
Recall	0.916
F-score	0.9095
Accuracy	0.9753

Table 2: Quantitative comparison of our results on SpaceNet6.

Performance metrics. We use mean IoU and accuracy to measure the performance of our experiments. The Intersection-Over-Union (IoU), also known as the Jaccard Index, calculates the overlap between the predicted segmentation mask and the ground truth divided by the union of prediction and ground truth.

SpaceNet6 dataset A great characteristic of GSCNN is that the shape branch can be removed at inference time since it has been designed to assist the main branch. We decided to study the effectiveness of GSCNN in a slightly different setting than traditional semantic segmentation. We pass SAR images to the main branch, but instead of extracting boundaries from SAR, we calculate image gradients from the RGB input image. This allows us to remove the shape branch at test time without hurting the performance. The problems that require multi-modal input are usually more difficult than their single modal setting. Figure 5 show the quality of our predictions on SpaceNet6. Table **??** provide quantitative results on the validation set.

3.4 Conclusion

In this chapter, we have presented a study on semantic segmentation for aerial imagery. However, these models cannot generalize to images from other cities, and as discussed before, collecting and annotating new data can be very expensive. For these reasons, we decided to design an unsupervised image-to-image translation network and generate synthetic data to fix this issue. We present our method in the next chapter.

Chapter 4

Image-to-Image Translation

This chapter is based on the conference paper "Unsupervised Structure-Consistent Image-to-Image Translation", Shahfar S., Poullis C., Under review.

Abstract

The Swapping Autoencoder architecture achieved state-of-the-art performance in deep image manipulation and image-to-image translation. In this work, we present a novel deep generative model for image manipulation. We introduce a new module that encourages better disentanglement between the structure and the style, based on gradient reversal layers. Furthermore, we present an attribute-based transfer method to achieve a more refined control in style transfer while preserving structural information without requiring a semantic mask. To manipulate an image, we encode both the geometry of the objects and the general style of the input images into two latent codes with an additional constraint that enforces structure consistency. The superiority of the proposed model is demonstrated on complex domains such as aerial images where existing state-of-the-art are known to fail. Moreover, our model improves the quality metrics for a wide range of datasets while achieving comparable results with multi-modal image generation techniques such as SMIS and BicycleGAN.

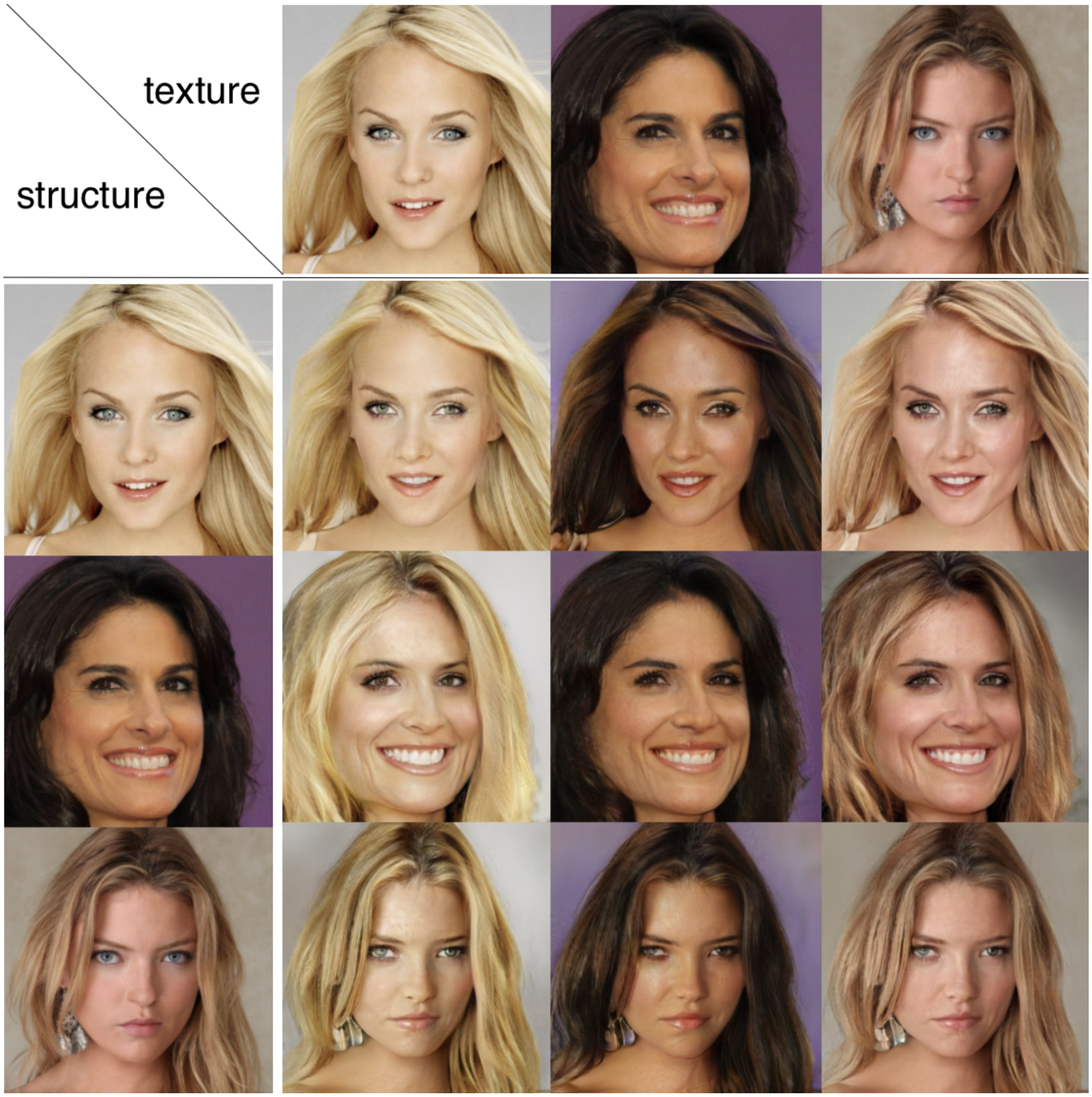

Figure 1: **Performance on CelebAMask-HQ:** Our model generates structure-consistent samples while transferring style from one image to another. Unlike most models that fail to preserve small structural details, our approach is able to preserve fine details such as earrings (see last row).

4.1 Introduction

Image-to-image translation and image manipulation techniques attracted much attention [15,31,36,38,40,53,55,84,89,98] recently as they can have a significant effect

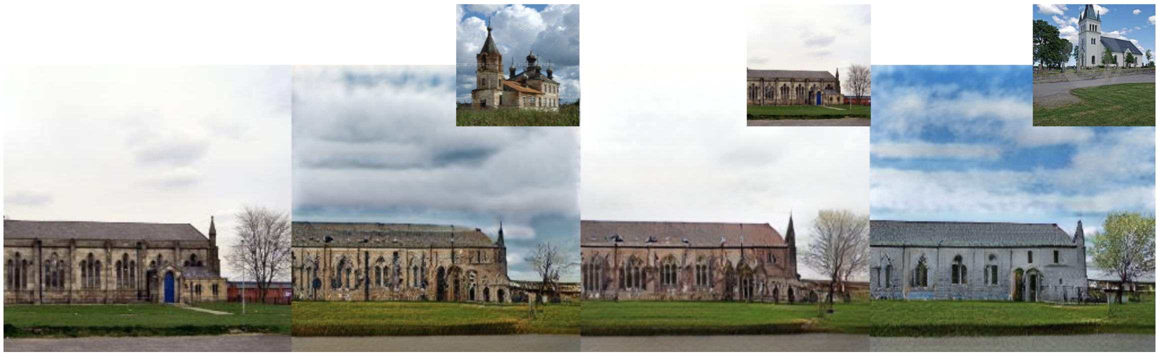

Figure 2: Our method learns structure-consistent image-to-image translation without requiring a semantic mask. We learn to disentangle structure and texture for applications such as style transfer and image editing tasks. The first(left) image shows the first input image, and the second/third/fourth images show the generated image where the structure is retained from the first input image and the texture from the second/third/fourth input image, which appear in the inset images. Note that the tree's structure is preserved, and its texture -in this case, the foliage's colour and density- changes according to the texture of the second input image in the inset. Our model was not trained on any season transfer dataset.

on many different tasks. Of particular interest is creating realistic synthetic training datasets to improve models' performance and generalization. One example that demonstrates the use of a synthetic dataset in the training of networks is presented in [96] where the authors introduce a semi-supervised approach to generate datasets for semantic segmentation.

There are a plethora of works [39, 40, 45] which report that for images containing single objects such as faces, or for images having the same semantic layout such as building facades, deep image manipulation techniques can produce realistic synthetic images. However, generating natural scenes or more visually complex images remains a challenge due to differences in the semantic layouts of the input images.

The challenge of deep image manipulation state-of-the-art with complex scenes is recognizing and learning essential features and characteristics from the input image. Structural information is typically shared or has common characteristics across different images in a dataset. On the other hand, the texture appears entangled with intrinsic image features. The standard approach to preserving the structural information is to condition the generation process on the input semantic mask using conditional image synthesis frameworks. However, that approach is not practical for image manipulation since the assumption of having access to semantic masks does not hold in most cases. Researchers explored different methods such as [53, 73], but in this work, we assume that image representations can be disentangled into the content/structure and texture/style.

To address this problem, we propose an auxiliary module that enforces the separation of structure from texture. This branch promotes the disentanglement of structure and texture by suppressing the existence of texture related information in structure code through applying gradient reversal layer. Additionally, it will encourage the emergence of deep features which is of high importance for image editing tasks. In this paper, we pursue three main objectives: 1) consistent and accurate structure preservation, 2) diverse, and 3) realistic image synthesis. Our goal is to learn multi-modal structure-consistent image-to-image translation in a fully unsupervised approach without requiring semantic segmentation masks.

Better structure preservation can impact many applications such as creating a 3D synthetic simulation world, image editing, semantic image synthesis and style transfer. To summarize, our technical contributions are:

- A new approach for a structure-consistent image-to-image translation that does not require previous knowledge on the geometry of the objects in the scene.
- An auxiliary module that enforces better disentanglement between the structure and texture information.
- An extension of the Swapping Autoencoder model with our auxiliary module.

We present experiments on several datasets, including the CelebAMaskHQ [48] Figure 1, the LSUN Church [90] Figure 2, and Cityscapes [12] Figure 6. Our results demonstrate that the proposed method improves the performance with less training.

4.2 Related Work

This section provides an overview of the most relevant state-of-the-art, grouped according to their methodology.

Generative models. Generative Adversarial Networks (GANs) [22] introduced an adversarial process to train a generative model. The problem is formulated as a zero-sum game between a generator and discriminator where the optimal solution is to find a Nash equilibrium. Ian J. Goodfellow refers to this framework as a minimax two-player game in which generator G tries to minimize the probability of the discriminator D to recognize the fake samples, and D tries to maximize the probability of assigning the correct label. The objective function can be written as follows:

$$\begin{aligned}\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] \\ + +E_{z\sim p_z(z)}[\log(1-D(G(z)))]\end{aligned} \tag{1}$$

GANs have proven to be very successful [7, 39, 40, 97] compared to other common approaches such as [30, 70, 80, 81, 83]. Both GANs and Variational Autoencoders (VAEs) [44] contain an encoder and a decoder; however, they differ in the sense that the GAN is a framework for estimating data distribution. On the other hand, VAEs learn the stochasticity within the data using the encoder's latent code to match the Gaussian distribution by reparameterizing the latent distribution and maximizing the log-likelihood function. Some methods [2, 99] combine GAN and VAE or GAN and Autoencoders in their models to achieve multi-modal image generation and prevent mode collapse.

Conditional generative models such as conditional VAEs [74], conditional GANs [62], conditional autoregressive methods [23,80], to name a few, have shown promising results [98] but we focus on conditional GANs for the rest of this section. Generative adversarial networks can be extended to conditional generative models [62] by feeding additional information c into the discriminator and generator. This c can be any information such as edge mask for semantic segmentation task or class labels for classification. By doing so, the generator can use prior noise $p_z(z)$ and additional information c to create a hidden representation and the discriminator will use the information provided as an input for a better discrimination. The quality of the results generated using conditional GANs inspired many applications employing this method, including, but not limited to, image-to-image translation [38,55,84,89], image editing [8, 27], image inpainting [56, 75, 88], text-to-image [87, 93], photo colorization [52, 71, 92, 95], conditional domain adaptation [4, 9, 10, 91], super resolution [37, 47], style transfer [17, 32, 37, 39, 40, 86]. Our work extends the image-to-image translation framework with a focus on image manipulation and style transfer.

Image-to-image translation is a framework to transfer an input image into a synthesized output image while preserving some information from the input. There are many methods designed for different applications. The main difference is in the information they preserve from the input image, which depends on the application. Image-to-image translation showed promise [15, 31, 36, 98], however, as stated in [99], the quality improvement may come with the cost of losing multi-modality. Recent works show that it is possible to prevent losing multi-modality and use this method for multi-domain scenarios [33, 49, 50, 99].

Unsupervised disentanglement aims to model the variations in data. It has been the focus of several pioneer works such as [7,29,73]. InfoGAN [7], for example, achieves this by maximizing the mutual information between latent variables and input data, whereas [41, 51, 66, 99] disentangle input information to structure and texture codes. Our work builds on the same principles to disentangle structure and texture in a completely unsupervised approach. However, we go one step further and aim for better disentanglement by introducing a new module to enforce better separation between the two. We show that our approach can achieve the desired disentanglement and generate realistic and diverse images while disentangling structure from style better than previous methods.

Multi-modal image synthesis overcomes the limitation of conditional GANs ignoring the latent code, also known as mode collapse. The idea behind the multi-modal image-to-image translation is to learn a conditional distribution while generating diverse images. Early works on conditional image-to-image translation were mostly focused on producing deterministic outputs [36, 55], which limits their applicability. In Section **??**, we show that our method can synthesize comparable results with the current state-of-the-art [99, 100].

Style transfer also known as texture transfer, can be defined as the problem of synthesizing an image with style extracted from the source image while preserving the semantics of the content image. Recent style transfer methods [39, 40] proposed the use of conditional normalization layers such as Conditional Instance Normalization [14] and Adaptive Instance Normalization [32] as a practical approach to transfer the global style. Normalization layers used in most style transfer methods are known to diminish semantic information. Spatially-Adaptive Normalization [65] was introduced as a way to avoid semantic-level information loss. We propose a closely related method for preserving semantic information without having access to a segmentation mask.

4.3 Method

Deep image manipulation requires an architecture with excellent feature extraction capabilities that allows for better disentanglement of texture from structure later on. Using an encoder, our goal is to disentangle the structure from the texture for both input images to our model. When swapping the texture or structure codes between the two randomly sampled input images $x_1, x_2 \in^{H \times W \times 3}$, our model can synthesize an image with the same structural information as to its content reference, but having the visual appearance or texture of the style reference image. Thus, we aim to generate realistic synthesized images where the structure for the first image is preserved while transferring the style from the second image.

Our solution comprises three key modules with two discriminators namely D and D_{style} as shown in Figure 3: an encoder E, a generator G, and a disentanglement module T which enforces better disentanglement of the structure from the style. The encoder learns how to encode visual information into two latent codes. Similar to [66], we enforce a mapping from any combination of the two latent codes to a realistic image by training an autoencoder. The generator is responsible for synthesizing realistic images using the two extracted latent codes. The disentanglement module is designed to enforce the separation of the structure from the texture. We present the details of the objective function in the subsequent sections.

4.3.1 Encoder

The encoder E learns a mapping from the input image to two latent codes corresponding to the structure and the texture. We use a traditional autoencoder training process. We employ a reconstruction loss to measure the difference between the original image and the synthesized version with an additional non-saturating adversarial loss [22] to enforce realistic image generation, and is defined as,

$$\begin{aligned} L_{enc}(x_1, \hat{x_1}) &= L_{rec}(E, G) + L_{adv}(E, G, D) \\ &= \|x_1 - G(E(x_1))\|_1 - \log(D(G(E(x_1)))) \end{aligned} \tag{2}$$

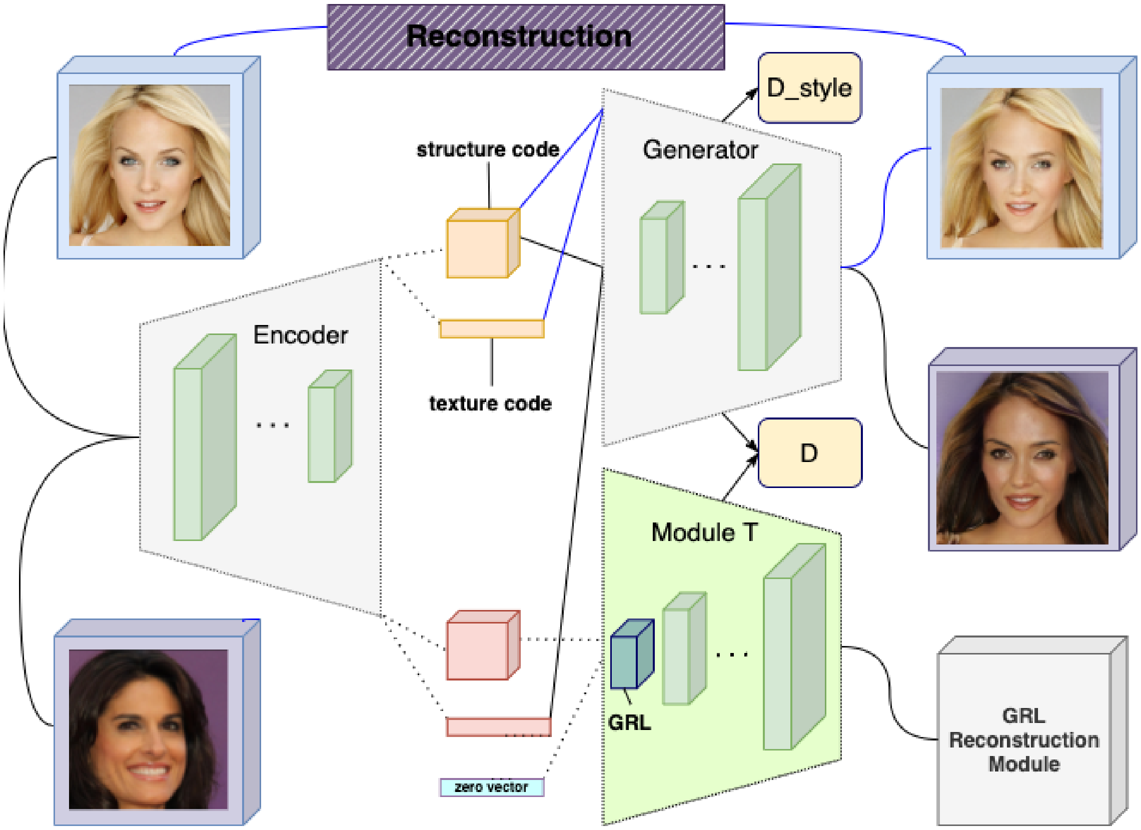

Figure 3: **Overview.** The geometry of the objects and the general style of the input images are encoded into two latent codes with an additional constraint that enforces structure consistency. We introduce a new module that encourages better disentanglement between the structure and the style, based on gradient reversal layers. This results in an attribute-based transfer that allows for a finer style transfer control while preserving structural information without requiring a semantic mask.

4.3.2 Generator

Assuming we have already learned how to disentangle the structure from the texture, we can pass two images x_1, x_2 to the encoder and get the latent codes z_1, z_2 where $z_1 = (z_s^1, z_t^1)$and $z_2 = (z_s^2, z_t^2)$. The generator conditioned on the latent structure code learns to map the extracted structure and texture codes to an image. The texture code will be added through weight modulation/demodulation introduced in [40]. Swapping the two texture codes before passing them to the generator is a common method to transfer style from one image to another. To ensure that the generated image is realistic, an additional non-saturating adversarial loss [22] is added to the overall objective function:

$$L_{swap}(E, G, D) = -\log(D(G(z_s^1, z_t^2))) \tag{3}$$

4.3.3 Structure and texture disentanglement

The latent codes must represent the structure and texture. However, this cannot be achieved in our current setting without additional constraints to encourage consistent structure and texture disentanglement. The approach used for learning consistent texture codes is to enforce all the patches sampled from the image generated in the previous step by swapping the textures to be visually similar to patches extracted from the texture reference image [66]. We achieve this using the following loss:

$$L_{style}(E, G, D_{style}) = -\log(D_{style}(C(G(z_s^1, z_t^2)), C(x^2)))) \tag{4}$$

where C is a random crop of size in the range $[\frac{1}{8}, \frac{1}{4}]$. This formulation results in learning a more consistent style transfer. Experiments have shown that this term is not enough and that better disentanglement can be achieved by enforcing the structure code not to contain texture-related information. In order to enforce structure consistency, we introduce an extra module with a gradient reversal layer (GRL) followed by a generator. This new generator has the same architecture as the original generator, but it reconstructs an image with an all-zero texture code that is theoretically impossible. Our analysis of previous works shows that structure code contains spatial information and includes style-related information. An inconsistent encoding will cause the network to generate odd samples that do not follow the algorithms and cannot be interpreted. We train this module using a reconstruction loss and a

Method	LSUN Church	#iterations
StyleGAN2 [40]	57.54	48 M
Swapping [66]	52.34	14 days x 4 V100 GPUs
Ours(validation)	52.96±1.7	4M

Table 1: Quantitative comparison of FID and training time/number of iterations on the validation set with state of the art methods. Our proposed method achieves comparable performance in terms of FID while it converge significantly faster.

non-saturating adversarial loss [22].

$$\begin{aligned} L_{aux}(x_1, \hat{x_1}) &= L_{rec}(E, T) + L_{adv}(E, T, D) \\ &= \|x_1 - T(E(x_1))\|_1 - \log(D(T(E(x_1)))) \end{aligned} \tag{5}$$

Adding the gradient reversal layer, as shown in [16], forces the encoder to suppress any style-related information in the structure code. It also proved to be useful in cross domain disentanglement [20]. The auxiliary loss from this branch would help the encoder to disentangle structure from texture better.

4.3.4 Objective function

We jointly train the encoder, generators and discriminators to optimize the final objective, which is the weighted sum of previously mentioned loss functions and is given by,

$$L_{total} = \lambda_{rec} L_{enc} + \lambda_{swap} L_{swap} + \lambda_{style} L_{style} + \lambda_{aux} L_{aux} \tag{6}$$

where $\lambda_{rec}, \lambda_{swap}, \lambda_{style}, \lambda_{aux}$ are weights that control the importance of each term. We will discuss the optimal values found for each term in section **??**.

4.4 Experiments

Implementation details. In all reported experiments, we randomly crop and resize the input images to 256 × 256 resolution. We use the Adam optimizer [42] with $\beta_1 = 0.0$, $\beta_2 = 0.99$. All reported results are computed on at least 4 NVIDIA TESLA P100 GPUs. The discriminator D is based on StyleGAN 2 [40] and D_{style} is based on Swapping autoencoder [66]. We experimented with different hyper-parameters for $\lambda_{rec}, \lambda_{swap}, \lambda_{style}, \lambda_{aux}$ but in this version we simply set the loss weights to be all 1.0.

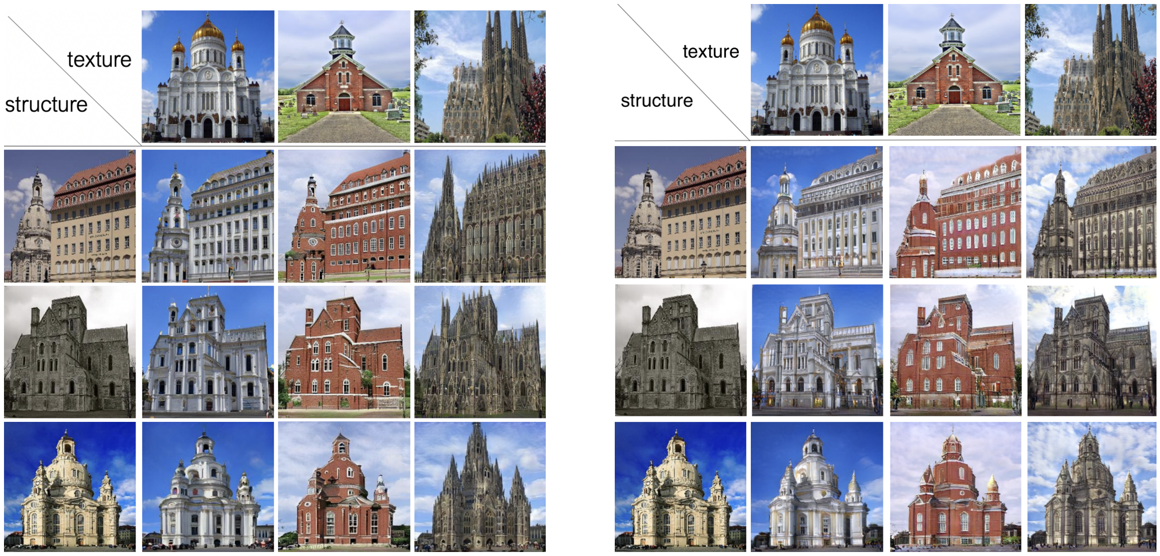

Figure 4: Left: Results from Swapping Autoencoder [66] on LSUN Church. Right: Our results on the same images. As evident, our model achieves better feature embedding and can retain the structural information of the input image while swapping only the texture with that of a second input image. Finer-level details such as spires and buildings outline are also retained. Most notably, our model was trained for a fraction of iterations compared to [66].

Datasets. We evaluate our method on the following benchmark datasets.

- CelebAMask-HQ [48] has 30,000 face images collected from the CelebA [58] dataset. CelebAMask-HQ contains annotations for 19 classes. However, we do not use masks in our training pipeline.

- LSUN church [90] is a subset of the Large-scale Scene Understanding (LSUN) dataset. The training set contains 126,227 images. It is a challenging dataset if no preprocessing is applied due to the diversity of the images.

- Cityscapes [12] is a street view dataset collected from 50 cities across Germany. The training set contains 3000 images with fine annotations, and the test set contains 500 images. It is considered a challenging dataset for image-to-image translation because each scene may contain up to 30 classes.

- Inria [61] is an aerial imagery dataset designed for semantic segmentation of building footprints. The training set contains 180 images with 5000 × 5000 resolution from 5 cities. Each image covers an area of approximately $1500m \times 1500m$. The test set contains 180 images of the same size collected from 5 cities that are not part of the training set.

Baselines. We compare our approach to a number of image-to-image translation, style transfer and multi-modal image synthesis methods including Swapping Autoencoder [66], StyleGAN2 [40], SMIS [100] and BicycleGAN [99]. We either use the results published by authors or generated using their official source code for all comparisons.

Performance metrics. We use Fréchet Inception Distance (FID) [28] to measure the quality of generated images. FID calculates the difference between the real and the generated data distributions using the Inception network to extract the features.

Structure-consistent style transfer. This section evaluates the quality of our generated images on style transfer and compares them to state-of-the-art. In Figure 4, we provide a qualitative comparison of our synthesized images with our baselines. We find that our method produces comparable results with [66] and [40] on LSUN Church dataset. A significant advantage of our approach is that it required only 3.9M iterations for training which demonstrates that not only is our approach significantly faster than our predecessors, but it surpasses their performance in terms of FID on the validation set, as shown in sota. Figure 4 shows that our method can generate samples with high visual quality on style transfer while preserving structure. Furthermore, structure similarity across generated samples supports the idea behind our auxiliary branch.

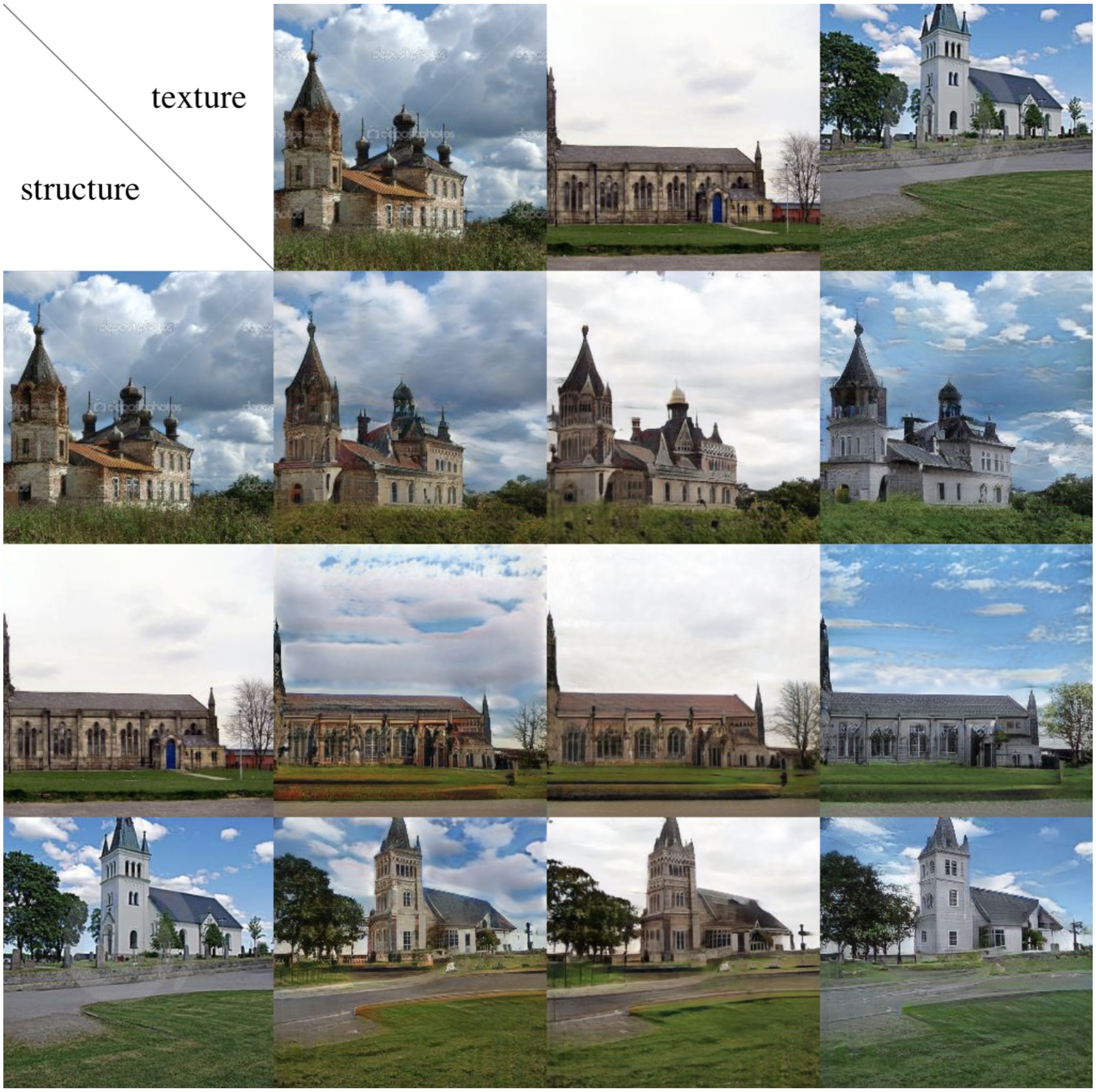

Figure 5: **Image translation on LSUN Church:** Each column corresponds to a particular texture extracted form the images on first row, respectively, each row contain the generated images with shared structure embedding.

Realism of reconstruction. The diagonals of Figure 1, 7 and 5 show the quality of our method on image reconstruction task from the learned feature embedding.

Our method preserves windows, doorways, trees, spires and generally the geometry of the objects as well as finer details such as earrings and tank top strap in Figure 1 (second row). We report quantitative comparison using the LPIPS [94] to compare the similarity of reconstructed images in the supplementary material.

Disentanglement of structure and texture. Accurately disentangling structure and texture is an important task both for style transfer and image manipulation. Given that this disentanglement is performed entirely unsupervised, we can evaluate the effectiveness of our new module by comparing the performance of our method with previous works on style transfer from existing images. Better disentanglement of structure and texture leads to a finer manipulation, resulting in significantly more realistic images. Figure 4 (left) shows the results from Swapping Autoencoder [66] on LSUN Church. Our results, shown on the right, demonstrate that our model achieves better feature embedding and generates images that retain the structural information of the input image while transferring only the texture from the second input image. Finer-level details such as spires and buildings' outlines are also preserved.

Texture code normalization. We evaluated the effect of normalization on the texture latent code and found that applying $\mathcal{L}_2$-norm results in faster convergence and more realistic synthesis. In this work we do not employ normalization in the generator, as in [34, 79], and similar to [66].

Applications. In Figure 5, we show examples from LSUN Church [90] that showcase the applicability of our method to other contexts. The bottom row shows a concrete example of how our technique preserves structures while transferring fine details. As it is evident, the building's structure is preserved while the texture is replaced. Similarly, the tree's structure is preserved, and its texture -in this case, the foliage's colour and density- changes according to each of the source images appearing in the top row. It should be noted that the model was not trained on any season transfer dataset.

Semantic image synthesis is one of the critical tasks in designing 3D environments, image colorization, and image editing, but it requires semantic masks and corresponding input images for training a model. This poses a limitation for many real-world applications where it is not simple to produce segmentation masks to train a conditional generative model in a supervised setting, but they need accurate semantic consistency. Our method can perfectly adopt for semantically multi-modal image

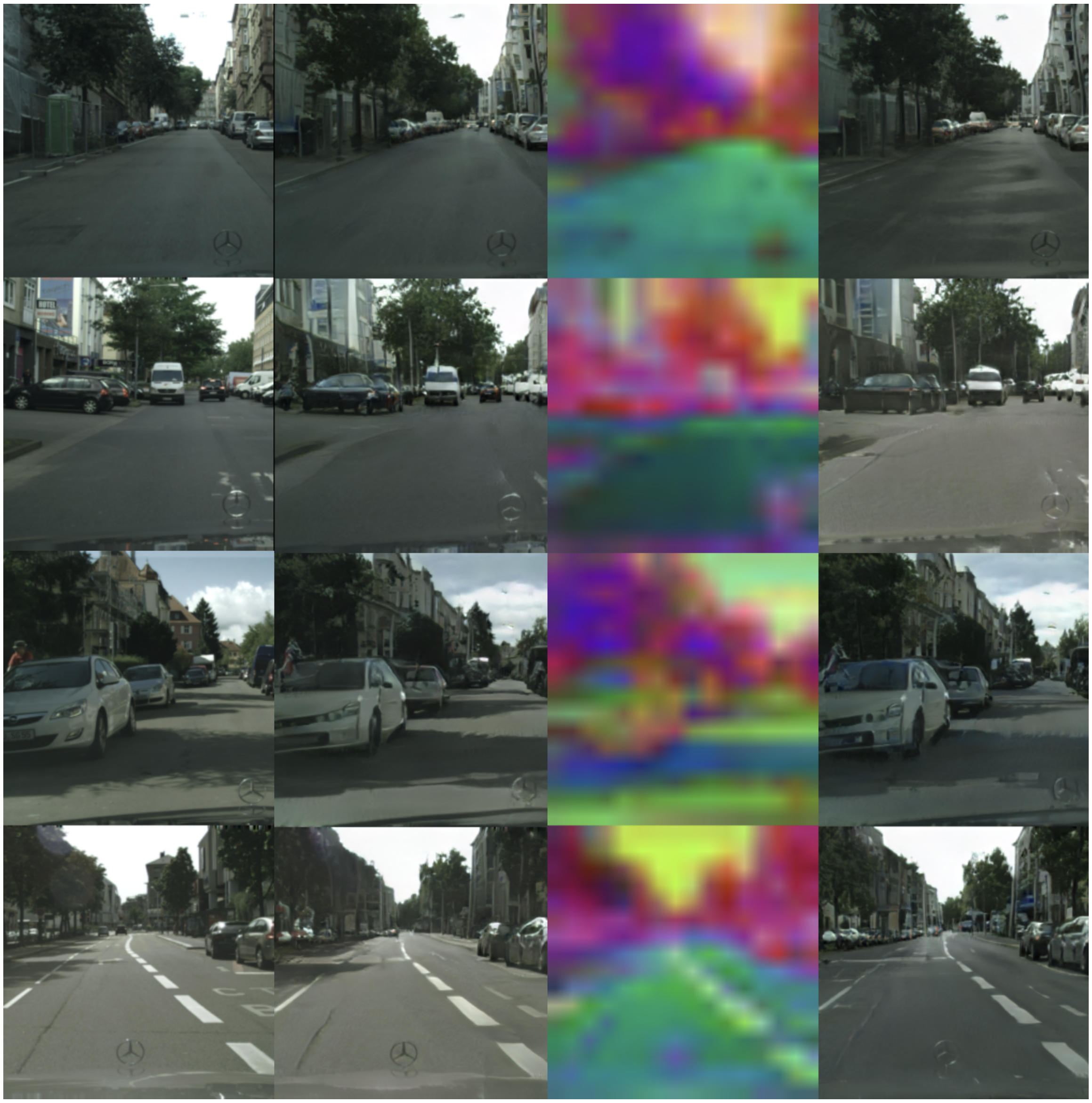

Figure 6: The left column shows the input images from Cityscapes, the second column are reconstruction of input images. We provide a visualization of structure latent codes in the third column after applying PCA and then resizing it to 256×256 for the purpose of visualization. The last column shows our generated images by swapping the texture between first and third row and between second and fourth row. As it can be seen the lightning information, asphalt texture and coloring of the facades are the main information that transferred by swapping the texture codes.

synthesis in an unsupervised setting.

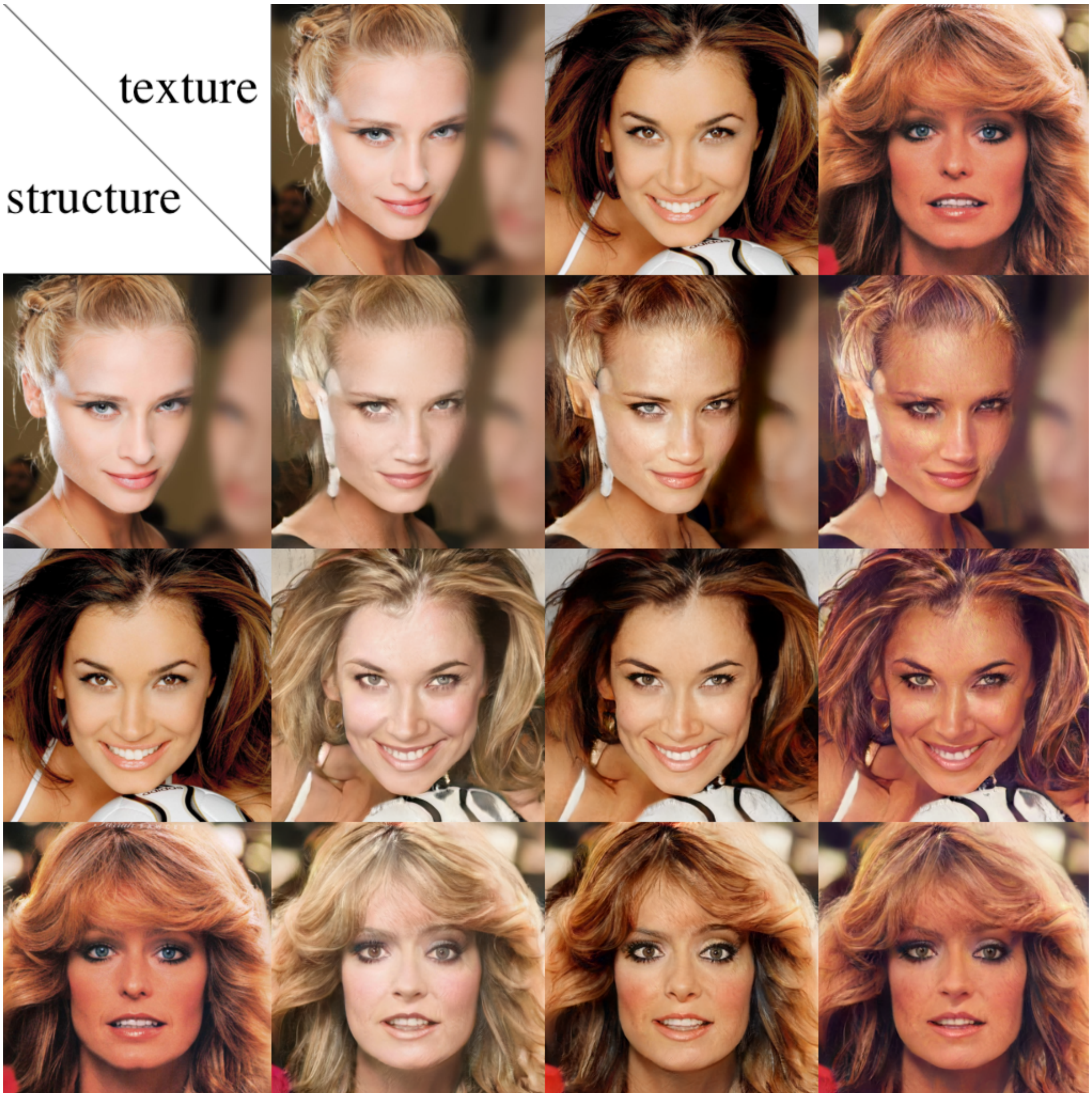

Figure 7: Qualitative comparison of style transfer on CelebAMask-HQ. In each column, the first row is the texture input image and the remaining rows are the outputs according to the structure input image (first image of each row). On the second row, we can see the specular highlight on the person's face is embedded as a structure and is retained.

Discussion and limitations. Our method is superior to state-of-the-art unsupervised approaches and gives comparable results to supervised techniques for image manipulation and image-to-image translation. We showed that incorporating the proposed auxiliary module as part of the training encourages better disentanglement of the structure from the texture and better feature embedding. This opens up new applications for image editing and style transfer. Previous works [13] explored the effect of combining multiple loss functions with different weights all in a single model using [29] to achieve better optimization. We believe this can be applied in the image manipulation setting, where the importance of structure versus texture may differ from one scenario to another by designing an architecture in which one can specify the percentage of structure versus texture for image generation.

4.5 Limitations

The proposed method works best when both structure reference and texture reference images contain the same class of objects and it struggle if you try to transfer style from an image that does not have that object. The behaviour of our model is not completely predictable in those cases. As you can see in the last row of Figure 8, where the network added trees on a position where it supposed to be a building, our network may generate an image very close to the structure image with very little change or it may replace some objects. However, we didn't remove such cases during training but ignoring them can be a reasonable next step for style transfer tasks unless we achieve a better understanding of underlying meaning of learned texture embedding.

4.6 Applications

The motivation of our work is to remove biases from training datasets caused by class imbalances. Many benchmark datasets introduce bias [12, 59] that can limit the generalization capability of any network trained on them. Moreover, it can significantly limit the impact of methods trained on these datasets in real-world scenarios.

In this section, we present two unique applications of our technique:

- The first application addresses bias in training datasets and demonstrates how our method contributes to overcoming this issue.
- The second application addresses the cost-effective generation of training datasets for the task of semantic segmentation in satellite images while incurring no additional labelling cost.

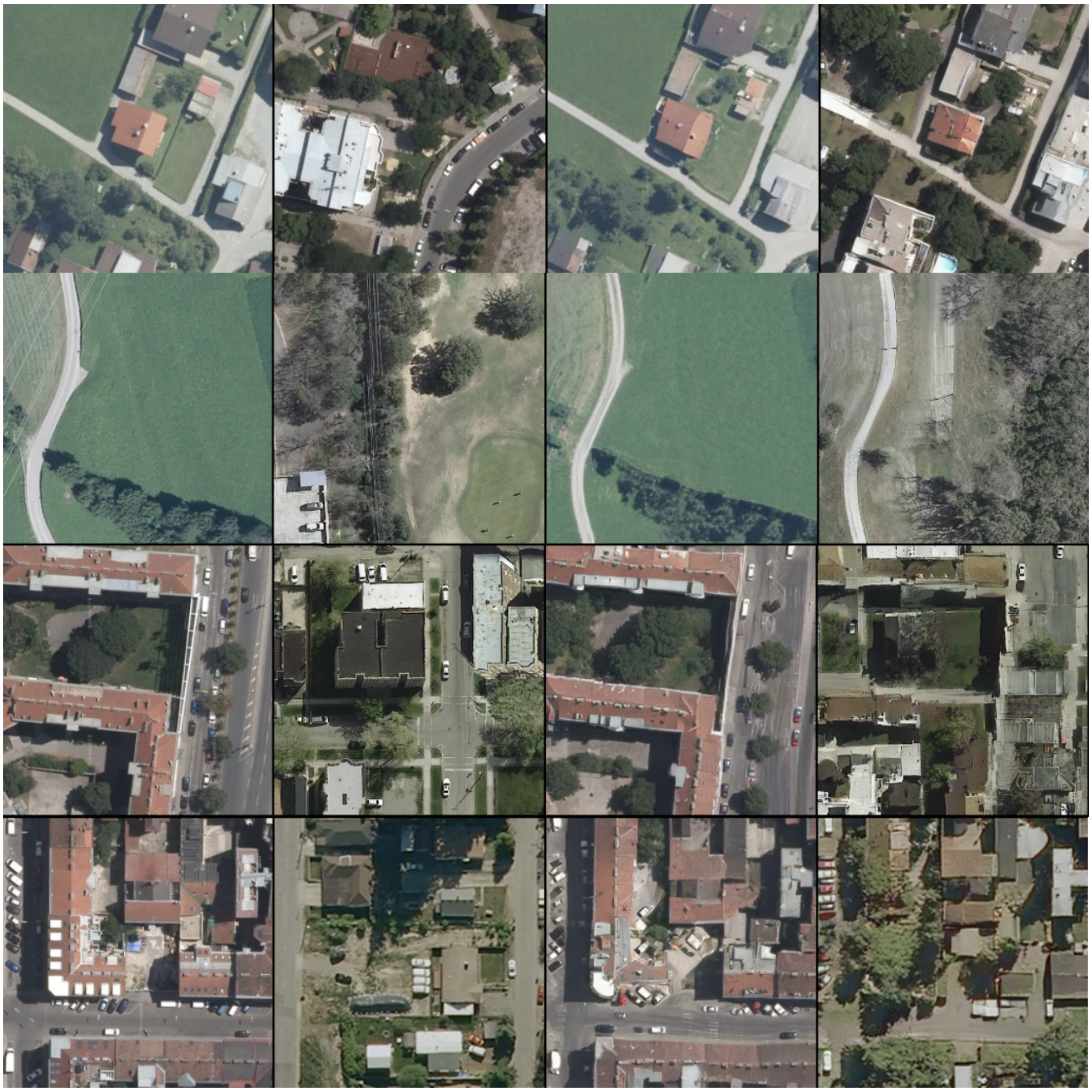

Figure 8: **Performance on Inria dataset.** Left-to-right: first input x_1, second input x_2, reconstruction of x_1, our generated sample using structure of x_1 and texture of x_2. The first column shows the structure reference images, the second column the texture reference images from which we extract their texture embedding. The third column shows the reconstructions of the first column using the learnt latent codes. The last column shows the synthesized images with the style transferred from the second image while preserving the structure of the first. Learning consistent embedding for aerial imagery is clearly more complex than single object datasets. We argue that given the complexity of the problem our method still preserves fine details such as front and back windows of the cars which shows its potential.

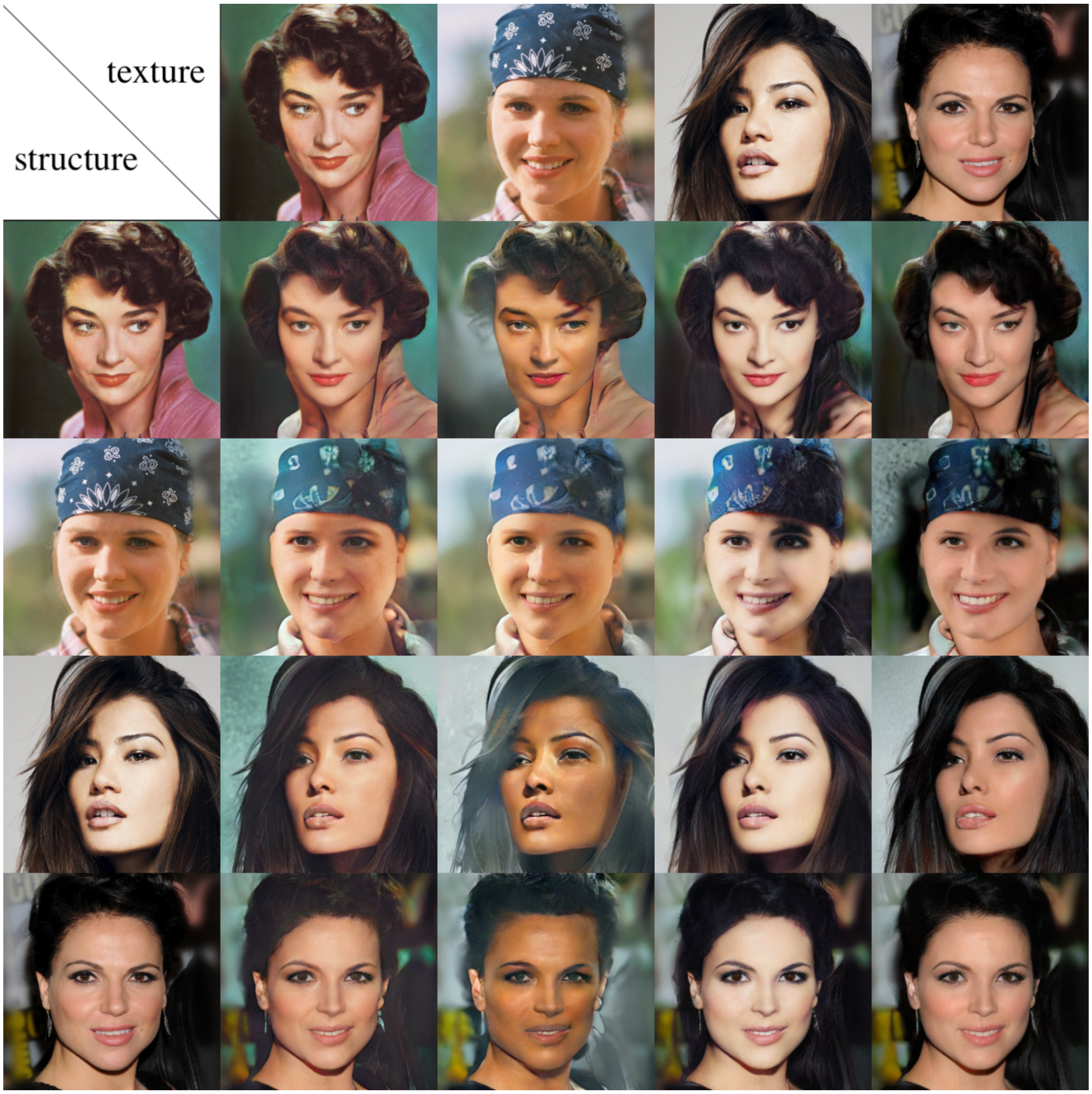

Figure 9: Additional qualitative results on CelebAMask-HQ dataset.

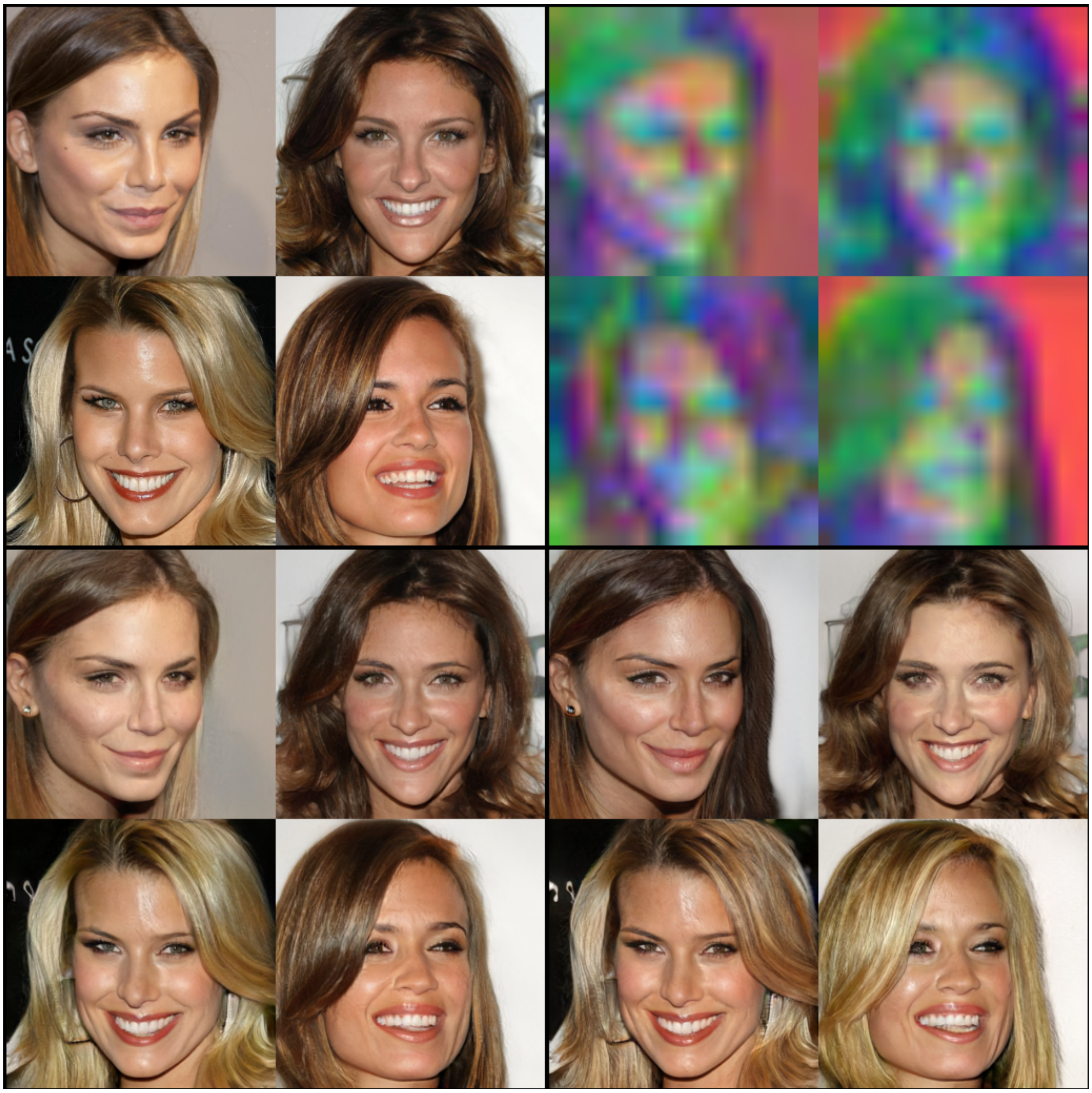

Figure 10: Illustration of structure embedding for a set of randomly sampled images. As shown in top right, the network learned to encode different body parts in the structure code.

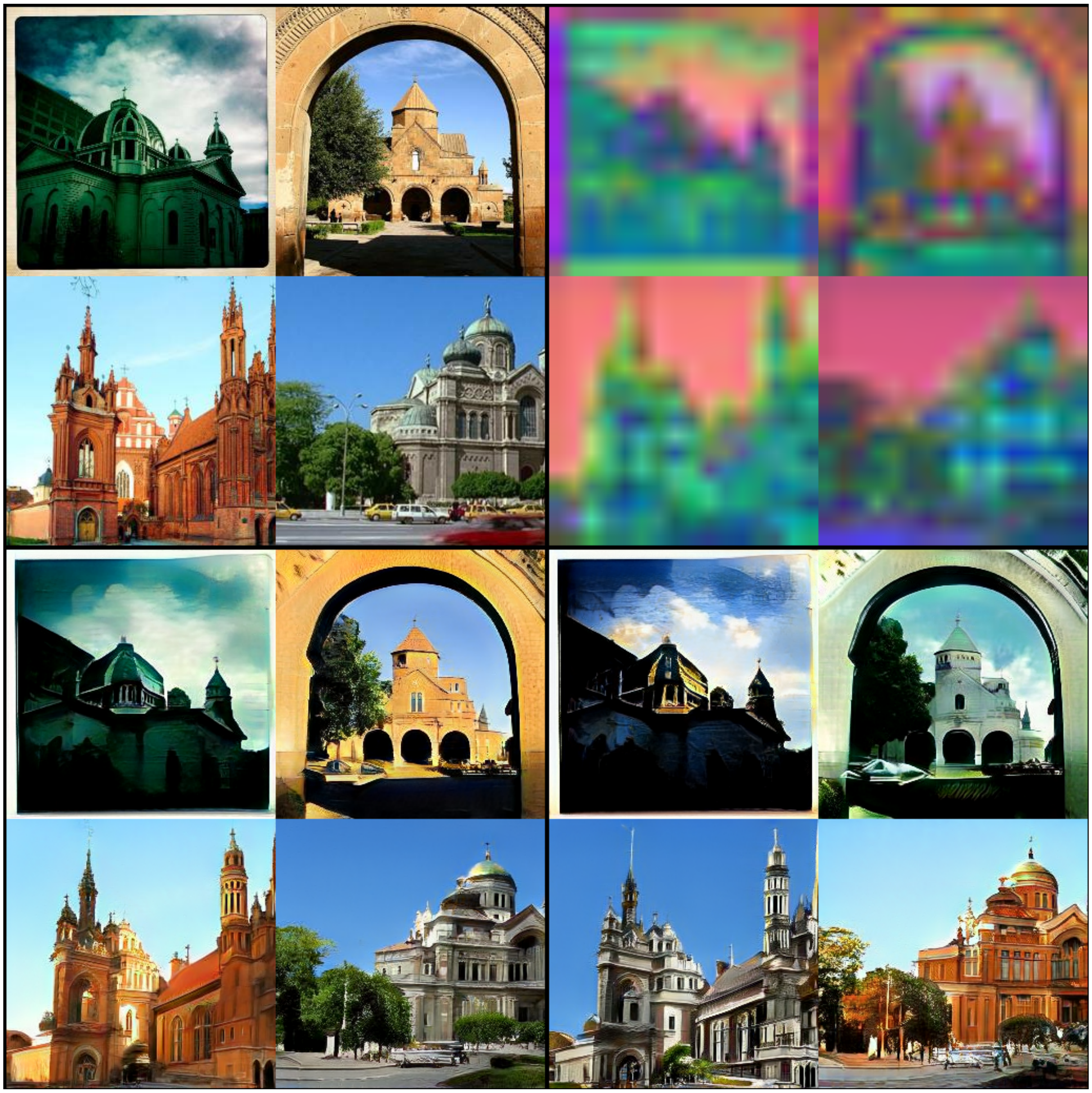

Figure 11: Illustration of some limitations of our method. Top right shows the learned structure embedding.

Furthermore, we present additional comparisons with state-of-the-art and quantitative results on the datasets LSUN Church [90], CelebAMask-HQ [48], Inria [61]. Finally, we conclude with a discussion on the limitations of our technique.

4.6.1 Application: Addressing bias in training datasets

Often we talk about biases in different datasets as an issue that needs to be addressed while designing the method, and we observe some generalization issues caused mainly due to imbalances in class distributions. A different approach is to adjust or expand our existing datasets to overcome this issue. Our method can preserve fine details; for example, in face datasets, these often imbalanced features can be gender, age, skin colour, hair colour, and accessories such as earrings, eyeglasses, hats, etc. Using our method allows us to balance the dataset by generating synthetic images with under-represented features. Furthermore, in cases where labels are available for the source image, these will also be the same for the generated images since our method preserves the same structure as the source image and only changes the appearance. Examples are shown in Figure 12.

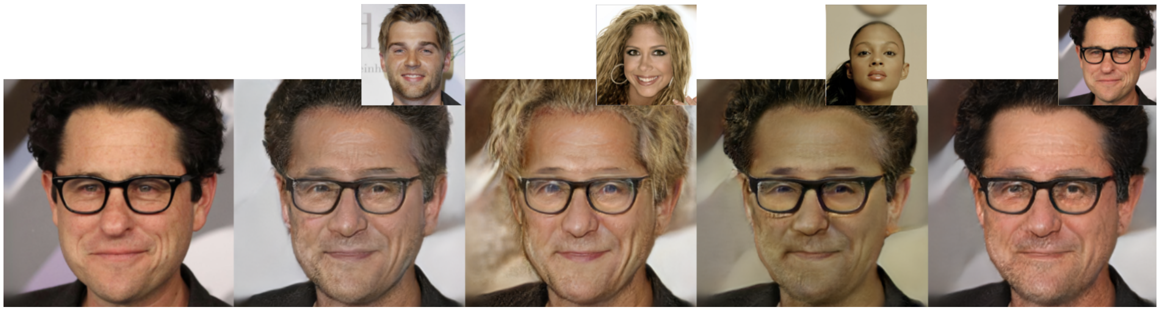

Figure 12: In all rows, the first(left) image shows the first input image, and the second/third/fourth images show the generated image where the structure is retained from the first input image and the texture from the second/third/fourth input image, which appear in the inset images.

4.6.2 Application: Training datasets for Semantic Segmentation of Satellite Images

Collecting satellite imagery for semantic segmentation is known to be an expensive and challenging task. The process of capturing images is expensive, but it may also contain inaccuracies due to the dynamic environment, e.g. a new building may appear that was not present at the time of acquisition of the satellite images. Another common issue is that the data collected from one city/continent cannot be easily generalized for a different city/continent. Considering all the challenges mentioned above, deploying a semantic segmentation network for aerial imagery can be challenging. Our structure-consistent network is designed to help overcome these challenges by generating realistic samples for different cities and weather conditions and generally creating datasets by style transfer. Our approach significantly reduces the time needed to process the data since we can expand any existing dataset to the desired style by only having a few images from the new city without requiring semantic labels Figure 15. Moreover, it can also be extremely useful for editing or expanding already existing datasets by changing the learned structure embedding.

4.7 Additional comparison to state-of-the-art

In Figure 13, 14, and 15, 16, 10, 4.5 we provide additional qualitative results on both reconstruction and style transfer tasks.

4.8 Quantitative results

Table 2 provides quantitative comparison of our method with Swapping Autoencoder [66], StyleGAN2 [40], MaskGAN [48], SMIS [100], and BicycleGAN [99].

4.9 Conclusion

We present an end-to-end process for training a structure-consistent image manipulation of existing images. We show that our approach can disentangle structure and texture with higher accuracy while preserving finer details than previous works. We

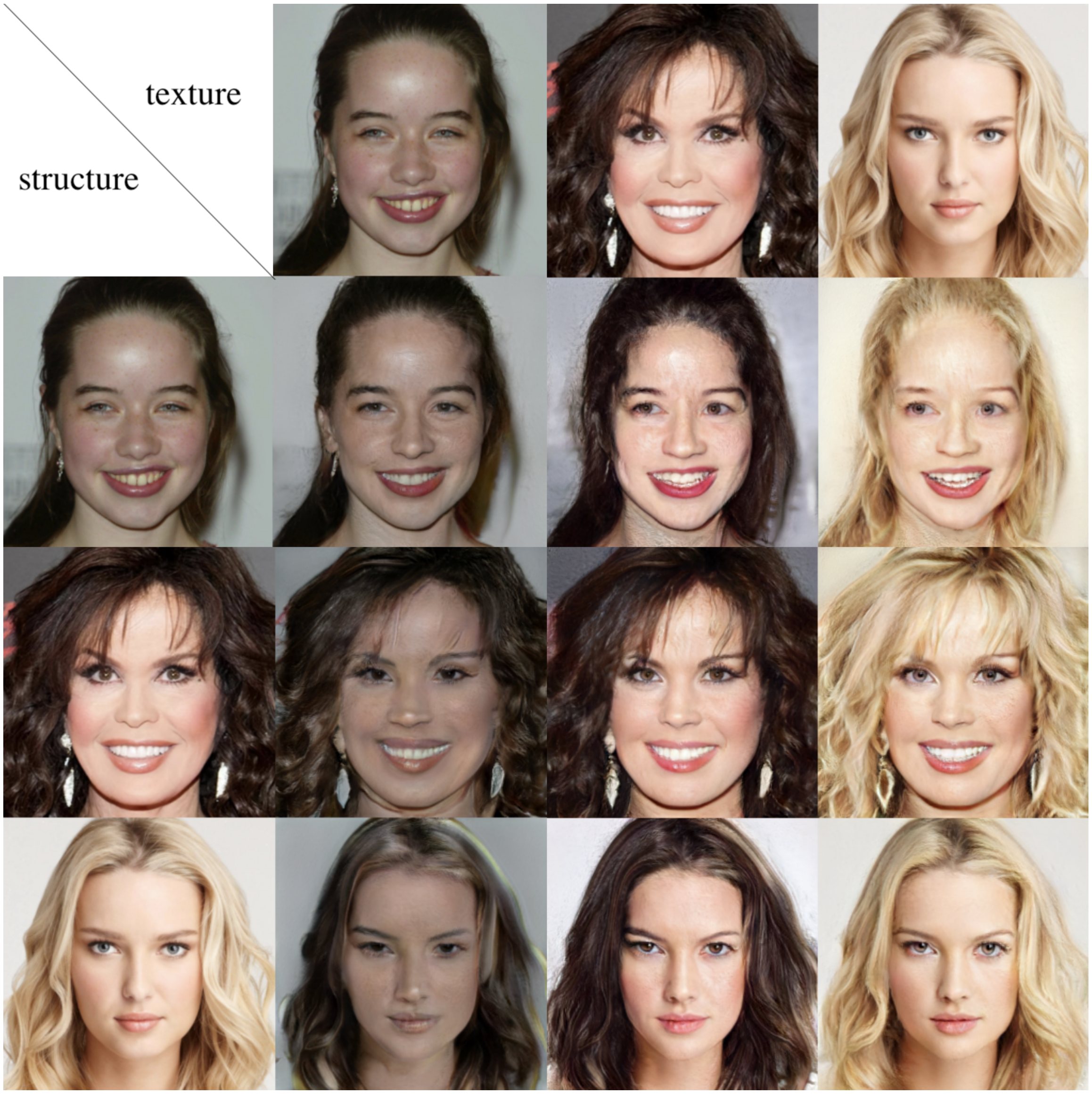

Figure 13: Illustration of style transfer on CelebAMask-HQ using our learned embedding.

have extensively tested our method and showed that it could consistently transfer texture to the correct parts and preserve structural information without requiring a semantic mask. Most notably, this is achieved while also reducing the computational time needed for training such a network. Although our method outperforms some state-of-the-art in the image-to-image translation task, defining and disentangling structure from texture in multi-object scenarios such as Cityscapes remains challenging due to the diversity of the objects and complexity of the scene. In the future,

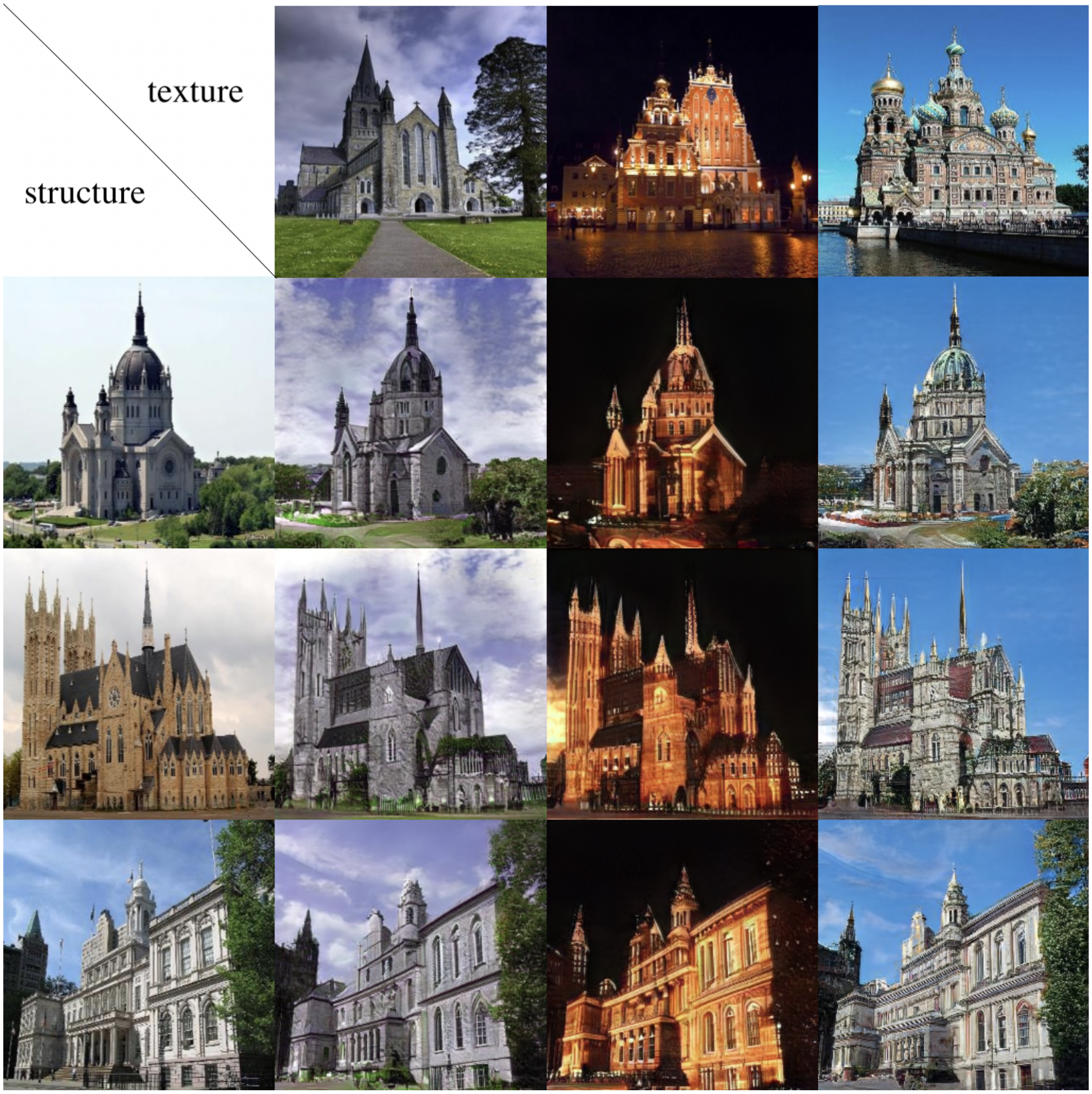

Figure 14: Image translation on LSUN Church showing the quality of our method in different lightning and weather.

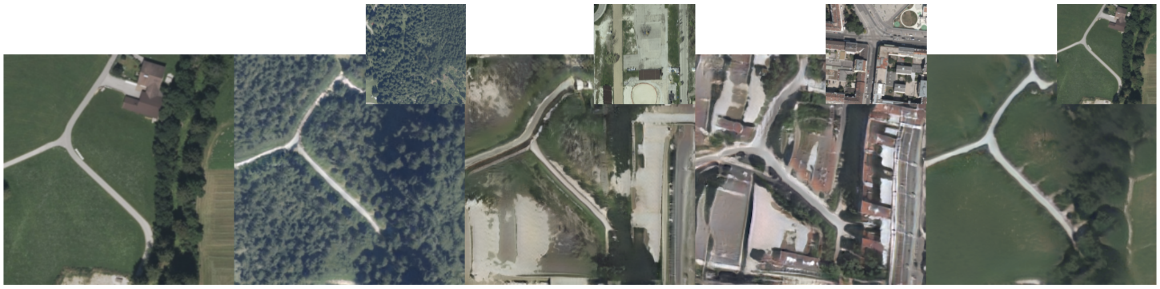

Figure 15: This figure provide an example of how our method can preserve the geometry of objects and semantic details while transferring the style. This would allow us to generate multiple samples with no extra labeling cost.

Method	LSUN Church	CelebAMask-HQ	Cityscapes
Ours	52.96±1.7	29.69	174.8
Swapping [66]	52.34	32.83	182.5
StyleGAN2 [40]	57.54	-	-
MaskGAN [48]	-	46.84	-
SMIS [100]	-	-	49.81
BicycleGAN [99]	-	-	87.74

Table 2: Quantitative comparison of FID on style transfer with some label-to-image translation work that are known for multimodal image synthesis and Swapping Autoencoder. In cases that we didn't have access to metric values calculated by the author, we trained their model for the same number of iterations as our network. Our method can achieve better results on CelebAMask-HQ and comparable results on LSUN Church trained for only 1.2M and 4M images.

we plan to explore the knowledge embedded in latent codes for different datasets and study extending this framework to other domains as discussed in Section 5.4.

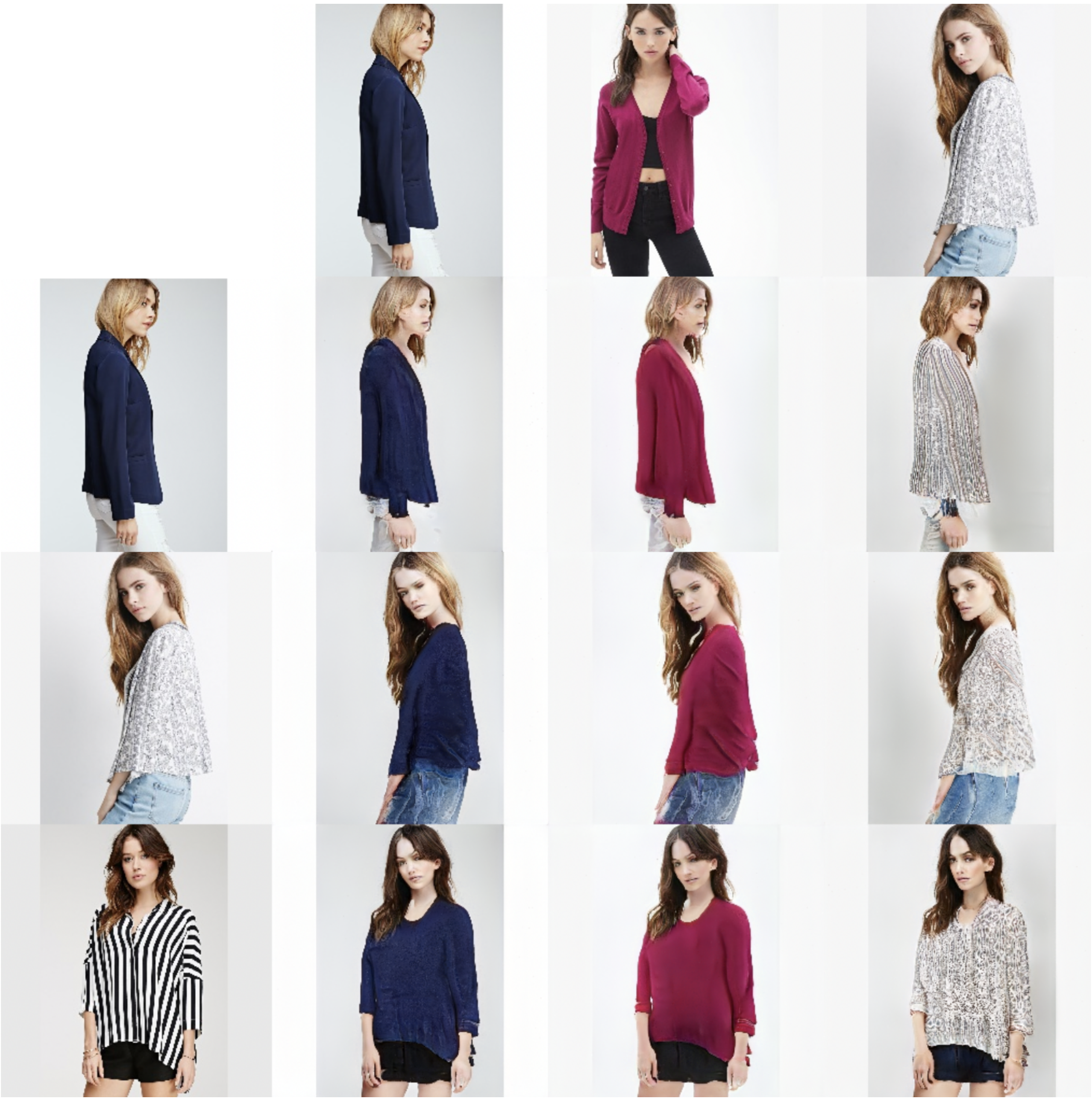

Figure 16: Results of our method on DeepFashion dataset. The model is trained and tested at 256px-by-256px.

Method	LSUN Church
StyleGAN2 [40]	0.377
Image2StyleGAN [1]	0.186
Swapping [66]	0.227
Ours	0.284

Table 3: Comparison of reconstructed image quality using LPIPS [94] on LSUN Church. Our method focus on preserving structural details and can produce high quality results. Given the fact that our model have only been trained on 4M images which reduce the training time by a great factor, our method can reconstruct input images better than StyleGAN2 [66].

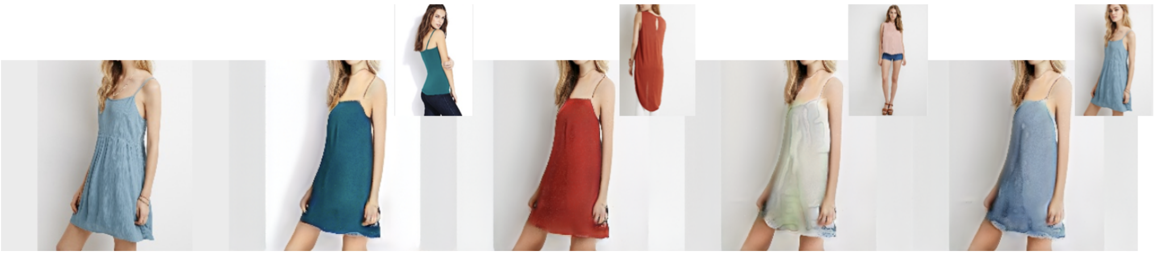

Figure 17: Results of our method on DeepFashion dataset. The model can accurately transfer the style of clothes and preserve the structure of the structure reference input.

Chapter 5

Conclusion

In this work, we introduced an unsupervised image-to-image translation method designed to create realistic synthetic datasets for domains that collecting data is costly. Our method improves the quality of image synthesis by generating structure-consistent images across different viewpoints. Our experiments on multiple datasets show improvements in disentangling structure and texture. Many recent works suggested that training on large datasets can significantly improve the performance of downstream tasks. However, it is not a reasonable next step in many domains due to environmental challenges and the high costs of creating such datasets. That is one of the main reason we focused on tackling this problem in this thesis and introduced our unsupervised structure-consistent image-to-image translation method. We, therefore, believe that our method can open up many doors for future work. Future directions could be as follows:

- Creating large datasets for Semantic Segmentation: The process of collecting data and creating annotations can be time consuming and expensive. However, [96] has demonstrated that it is possible to generate datasets for semantic segmentation by annotating a small fraction of synthesized images. Given that our method preserves structure, one can use few samples from existing datasets to train a segmentation module and to generate the annotations.

- Reducing bias in datasets: One of the sensitive challenges in data collection is to ensure that the dataset is not biased and that it is a fair representative of the population. Many existing semantic segmentation datasets suffer from bias that is caused by imbalances in class distributions. Our structure-consistent image generation network seems like a promising approach to balance the distributions by generating samples for under-represented classes.

- Image editing by exploring latent space: Given that we can separate the structure from the texture, exploring the structure and texture latent spaces by interpolating between two samples would be an interesting next step that can result in finding feature directions for image editing.
- Texture and structure interpretation: Studying interpretability of texture and structure embedding can guide researchers to achieve finer control in style transfer and image editing.

Bibliography

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvaegan: fine-grained image generation through asymmetric training. In *IEEE/CVF CVPR*, pages 2745–2754, 2017.

[3] Yoshua Bengio, Aaron C. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2013.

[4] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE/CVF CVPR*, July 2017.

[5] Bodhiswatta Chatterjee and Charalambos Poullis. On building classification from remote sensor imagery using deep neural networks and the relation between classification and reconstruction accuracy using border localization as proxy. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 41–48, 2019.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE CVF/CVPR*, June 2018.

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE/CVF CVPR*, pages 8789–8797, 2018.

[10] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF CVPR*, June 2020.

[11] Djork-Arné Clevert, Thomas Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv: Learning*, 2016.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF CVPR*, June 2016.

[13] Alexey Dosovitskiy and Josip Djolonga. You only train once: Loss-conditional training of deep networks. In *International Conference on Learning Representations*, 2020.

[14] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *CoRR*, abs/1610.07629, 2016.

[15] Patrick Esser, Johannes Haux, and Bjorn Ommer. Unsupervised robust disentangling of latent characteristics for image synthesis. In *IEEE/CVF CVPR*, pages 2699–2709, 2019.

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[17] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *2016 IEEE CVF/CVPR*, pages 2414–2423, 2016.

[18] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[19] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[20] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. *arXiv preprint arXiv:1805.09730*, 2018.

[21] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. Deep learning. *Nature*, 521:436–444, 2015.

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

[23] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.

[24] J. Han and C. Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *IWANN*, 1995.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[27] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[29] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[30] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[31] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. Disentangling factors of variation by mixing them. In *IEEE/CVF CVPR*, pages 3399–3407, 2018.

[32] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, Oct 2017.

[33] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[34] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[35] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[36] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE/CVF CVPR*, pages 1125–1134, 2017.

[37] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[38] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. Generative attribute controller with conditional filtered generative adversarial networks. *2017 IEEE CVF/CVPR*, pages 7006–7015, 2017.

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF CVPR*, pages 4401–4410, 2019.

[40] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE CVF/CVPR*, pages 8110–8119, 2020.

[41] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. In *2019 IEEE WACV*, pages 848–856. IEEE, 2019.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[44] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[45] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *ICCV*, October 2019.

[46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[47] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan

Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE/CVF CVPR*, pages 4681–4690, 2017.

[48] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE CVF/CVPR*, 2020.

[49] Hsin-Ying Lee, , Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[50] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018.

[51] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018.

[52] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *IEEE/CVF CVPR*, June 2020.

[53] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. In *IEEE/CVF CVPR*, June 2020.

[54] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

[55] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[56] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE/CVF CVPR*, pages 10551–10560, 2019.

[57] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[58] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, December 2015.

[59] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[60] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

[61] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017.

[62] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[63] V. Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[64] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[65] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE CVF/CVPR*, 2019.

[66] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, 2020.

[67] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[68] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[69] D. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[70] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[71] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE/CVF CVPR*, July 2017.

[72] Jacob Shermeyer, Daniel Hogan, Jason Brown, Adam Van Etten, Nicholas Weir, Fabio Pacifici, Ronny Hansch, Alexei Bastidas, Scott Soenen, Todd Bacastow, and Ryan Lewis. Spacenet 6: Multi-sensor all weather mapping dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[73] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *IEEE/CVF CVPR*, pages 6490–6499, 2019.

[74] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[75] Yuhang Song, Chao Yang, Zhe Lin, Hao Li, Qin Huang, and C-C Jay Kuo. Image inpainting using multi-scale feature image translation. *arXiv preprint arXiv:1711.08590*, 2:1, 2017.

[76] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[77] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1139–III–1147. JMLR.org, 2013.

[78] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[79] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

[80] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

[81] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[82] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1992.

[83] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[84] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE CVF/CVPR*, 2018.

[85] Wikipedia contributors. Loss function — Wikipedia, the free encyclopedia, 2021. [Online; accessed 27-March-2021].

[86] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE CVF/CVPR*, pages 8456–8465, 2018.

[87] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE/CVF CVPR*, June 2018.

[88] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2(3), 2016.

[89] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, Oct 2017.

[90] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[91] Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *Advances in Neural Information Processing Systems*, 2019.

[92] Bo Zhang, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar-based video colorization. June 2019.

[93] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, Oct 2017.

[94] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF CVPR*, June 2018.

[95] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

[96] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

[97] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, pages 597–613. Springer, 2016.

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

[99] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Multimodal image-to-image translation by enforcing bi-cycle consistency. In *Advances in neural information processing systems*, pages 465–476, 2017.

[100] Zhen Zhu, Zhiliang Xu, Ansheng You, and Xiang Bai. Semantically multi-modal image synthesis. In *IEEE/CVF CVPR*, June 2020.