Objects as Semantics in Style Transfer

Henry Allen

Bachelor of Science in Computer Science

The University of Bath

2022-2023

Objects as Semantics in Style Transfer

Submitted by: Henry Allen

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Neural Style Transfer (NST) is an active research area within non-photorealistic rendering, allowing a desired artistic style to be replicated from a style exemplar onto a target content image. Though NST imposes no restrictions on the semantic alignment of content-style image pairs, it is often the case that a more aesthetically pleasing image can be produced when the semantics align and are preserved during the transfer.

In this work, a semantic style transfer (SST) pipeline is proposed which is capable of automatically preserving local features in style images during the transfer process via spatial control, without the need for manual masking. Recent developments in large segmentation models [25] allow the domain of illustrative artwork to be sufficiently well-segmented, automatically, into coarse to semi-fine features: or generally speaking, into the concept of distinct 'objects'. Using such a segmentation of the style image as a base, non-parametric warping [39] can be used to build corresponding semantic segmentations between the content and style images, allowing for spatially guided SST methods to be employed. The pipeline is flexible, allowing any SST method to be used so long as it operates on semantic segmentations.

Contents

List of Figures

Acknowledgements

Thank you to my family and dear friends, who have always kept me in high spirits during the completion of this project. Thank you to my supervisor James Davenport for your guidance over the course of this project and proofreading of this report, and Peter Hall for the original project inspiration, and helping me gather my ideas. It would not have been possible without you.

Full-size PDF Note

For an uncompressed version of this PDF (>100MB), please visit Google Drive here (link working as of 2023/05/05). It may be useful for your reading if certain images in the compressed version are too compressed or low-resolution.

1 Introduction

Non-photorealistic rendering (NPR) is a sub-field at the intersection of computer graphics and computer vision in which the aim is to produce images which are intentionally detached from, yet inspired by, reality and photo-real images, such as those captured by photography. This is often achieved by generating or mimicking elements of artistic style or expressionism with complex algorithms. NPR techniques have a wide reach, spanning creative industries like animation, video games, and design, as well as scientific industries such as biomedical imaging and architectural visualisation.

Recently, the field of non-photorealistic rendering has been heavily influenced by artificial intelligence (AI), providing the impetus for a major paradigm shift. Nearly all new techniques leverage the computational capabilities of AI models to some extent. A particularly notable example is recent developments in large transformer models [50], which has pushed NPR even further into the limelight. Generative techniques such as Generative Adversarial Networks (GANs) [14] and Latent Diffusion models [47] have become very trendy as they offer a wide range of immediately appealing results, some of which are near indistinguishable from human creations at a glance.

Outside of generative methods, there exists techniques for manipulating existing images in an artistic fashion. Image to Image Translation [20] is a technique wherein an image is taken from a certain visual domain and translated into another, changing the depiction of the source material. A "domain" describes a set of images which share similar characteristics. The exact domains targeted vary depending on which are deemed useful to translate between; example domains could be on meta-characteristics such as colour vs. no colour, or on more nuanced characteristics such as artistic style in the works of different artists.

1.1 Style Transfer

Figure 1: In this paper, an automation pipeline is proposed for segmentation-based semantic style transfer methods. The pipeline has a style feature preserving focus, allowing intricate styles to be more accurately transferred than other methods when the semantics align. This figure displays results of style transfer to come. In each subfigure: content image [30] (left), style exemplar [64, 63] (middle), and style transfer output (right).

A similar yet disjoint technique, and the focus of this paper, is style transfer. Style transfer is a technique where the artistic style of an exemplar is imposed onto the content of another image, creating a stylised depiction of the source material. Originally, this was viewed as a texture-transfer problem, and CNNs were leveraged by the pioneering paper by Gatys *et al.* (2016) to extract style information from an exemplar and impose it onto another image via optimisation, birthing Neural Style Transfer (NST).

Developments have shown that texture transfer alone is an oversimplification of the problem of expressing artistic style. Though much research still looks at improving standard procedure in specific domains, an interesting avenue for development has been transfer through more true-to-form style depiction methods. For example, stroke-based rendering [26] has been employed to better approximate painterly styles, semantic segmentations have been used to better preserve artistic composition [13, 2, 52], and geometric deformation has been used to better approximate abstract and expressive styles [41, 39]. Methods which use this kind of additional semantic information fall under a subcategory of NST: Semantic Style Transfer (SST).

The recent success of these methods highlight the importance of auxiliary semantic information in the transfer process. This paper aims to contribute by making some of these methods more accessible, proposing an automated pipeline for segmentation-based Semantic Style Transfer methods. Additionally, particular cases are explored where the pipeline may perform better than current methods, specifically with regards to preserving local features within style images during the transfer process.

1.2 Arcimboldo: An Example

Figure 2: Some of Arcimboldo's most famous works. From left: Vertumnus [64], Four Seasons [63], Summer [66], Fire [61].

Giuseppe Arcimboldo was an influential painter in 16th century Italy who is best known for his imaginative and often surreal portraits (ahead of his time!). Arcimboldo's works present an interesting and challenging case for style transfer; his artistic style is specifically characterised by the composition of many local features. Each feature (typically, an object) could be described as having its own distinct visual 'style'. Additionally, the works have meticulous attention to detail. Thus, in order to capture the style properly, local style features both coarse and fine must be accurately preserved in large numbers, and texture transfer must be sufficiently well-detailed, all while geometrically aligning the feature composition to match the semantics of the content image.

Due to the challenge they propose for style transfer, Arcimboldo's works are used frequently in this paper as a high-level performance benchmark, and so are introduced here to give context to results which follow. Moreover, extensions of the proposed technique (section 4) are explored in an attempt to produce even better results for the specific domain of Arcimboldo's works.

1.3 Adapting to New Research Findings

The field of computer vision is a fast moving one, and even more so NPR. With the advent of aforementioned large transformer-based models [50], progress has begun moving at a blisteringly fast rate. Recently for example, style transfer has seen direct attempts at incorporating transformer models to better perform the transfer [8]. Significant developments within computer vision appear month-on-month, and can change the research landscape quickly.

This work is no exception. At the start of its timeline, segmentation of artwork images proved to be too challenging to enable a general pipeline like the final version presented in this work, given the performance of state-of-the-art methods. Even specific use cases like segmenting Arcimboldo's works were not solvable with out-of-the-box methods. As a result, research was conducted into alternative methods which could model specific features in artwork to produce a segmentation.

The initial plan was to choose a specific artwork domain - Arcimboldo's works - and model local features like fruit, vegetables, and flowers via models such as a point distribution model (PDM) [5]. From this, a segmentation could be produced, completing a similar segmentation-based SST pipeline to the final version you see in this work, albeit with less generality.

However, in response to significant recent research in semantic segmentation (See 2.3), a more general pipeline was able to be produced, somewhat changing the course of the project in certain areas. Notably, sections 2.3 and 3.2, which describe, and benefit from results in recent research.

2 Literature, Technology, and Data Review

2.1 Neural Style Transfer (NST)

Figure 3: Neural Style Transfer: a content image (left) [68] depicted in the artistic style of a style exemplar (middle) [74], and the resulting transfer output (right).

The seminal paper by Gatys *et al.* (2016) [12] outlines a technique to synthesize novel images by combining the content of one image: a content exemplar, with another image: a style exemplar. The technique leverages pre-trained deep convolutional neural networks (CNNs) to extract meaningful feature representations of content and style from the images. The 'content' feature vector used is typically pulled from a low-depth convolutional layer, in other words, some coarse understanding of the layout of the content image. On the other hand, the style feature vector is typically pulled from a range of low-to-high depth layers, and is analogous to understanding the texture build-up of the images. Originally, features were extracted from the VGG-16 network [48], but the newer VGG-19 network [48] offers deeper and more informative feature representations at the cost of processing time.

The style and content feature representations can then be used to generate a new image which preserves the content of one image and the style of the other via pixel-based optimization. This is achieved by defining two primary loss functions over each image: a content loss $\mathcal{L}_C$ and a style loss $\mathcal{L}_S$.

$$\mathcal{L}_C = \frac{1}{N_C} \sum_{\ell}^{N_C} \left(\mathbf{F}_{C_\ell}(\hat{\mathbf{x}}) - \mathbf{F}_{C_\ell}(\mathbf{x}_C) \right)^2 \tag{1}$$

$$\mathcal{L}_S = \frac{1}{N_S} \sum_{\ell}^{N_S} \left(\mathbf{G}\{\mathbf{F}_{S_\ell}(\hat{\mathbf{x}})\} - \mathbf{G}\{\mathbf{F}_{S_\ell}(\mathbf{x}_S)\} \right)^2 \tag{2}$$

where $\hat{\mathbf{x}}$ is the guidance image, the generated image at a particular iteration i. $\mathbf{x}_C$, $\mathbf{x}_S$ are the content and style images respectively. And N_C, N_S are the number of content and style layers used for the content and style feature vectors, respectively (typically $N_C = 1$, $N_S = 5$ when VGG networks are used). Note, it is also advisable to normalise feature vectors.

$\mathcal{L}_C$ is simply defined as the sum of mean-squared errors (MSE) between corresponding content features, layer-by-layer. $\mathcal{L}_S$, however, has gram matrices applied to style features (Equation 3). The result is a matrix which describes the correlations between style features across spatial locations in the images. A side effect of this is that the texture transfer process is spatially fixed. The significance of this observation will become apparent a little later.

$$\mathbf{G}\{\mathbf{F}\} = F^T F \tag{3}$$

Additionally, a supplementary loss function, total variation loss ($\mathcal{L}_{TV}$), is included which aims to impose cross-image coherence - spatial continuity in small pixel neighbourhoods. The result is more coherent texture in style transfer outputs, and the suppression of potential high-frequency generation artifacts such as colour banding or checker boarding. It is defined in Equation 4 as the mean squared differences of pixels which neighbour one another horizontally or vertically.

$$\mathcal{L}_{TV} = \frac{1}{W \cdot H}(tv_h + tv_v) \tag{4}$$

where

$$tv_h = \sum_{i,j}(x_{i+1,j} - x_{i,j})^2, \quad tv_v = \sum_{i,j}(x_{i,j+1} - x_{i,j})^2$$

Thus, the total loss is defined as

$$\mathcal{L}_{total} = \alpha\mathcal{L}_C + \beta\mathcal{L}_S + \gamma\mathcal{L}_{TV} \tag{5}$$

where $\alpha,\ \beta,\ \gamma$ are weighting constants chosen depending on which characteristics we wish to amplify.

2.2 Semantic Style Transfer (SST)

Semantic Style Transfer refers to derivative methods of the original NST paradigm, in which some auxiliary information is provided about the sources images, allowing for specific characteristics to be better represented in the style-transferred image. Typically, the semantic information being preserved is some characteristic of the content image, such as edges [3], salience [42], or semantic segmentation [13, 2, 59], among others [58, 40, 41].

In this work, the typical approach is flipped on its head, and instead the focus is preserving semantics from the *style* image by mapping them onto the content image. More specifically, by using semantic segmentation as first introduced by Gatys et al. (2017) [13], local style can be captured from specific regions in the style image. As presented, this process is not automatic however, and manual segmentation masks must be created [13, 2] to facilitate the process. The idea of automatically creating segmentation masks has been explored before [59], but the resulting segmentation is only partial and would not be sufficient for capturing fine details in a style image. Alternatively, recent methods opt for a *context aware* strategy where extracted features are automatically aligned in an attempt to transfer only between those that are similar [52, 34]. However, the alignment tends to be fairly coarse meaning some features are still missed, and furthermore manually-created coarse semantic segmentations are still required to guide the process.

2.2.1 Spatial Control via Guidance Channels

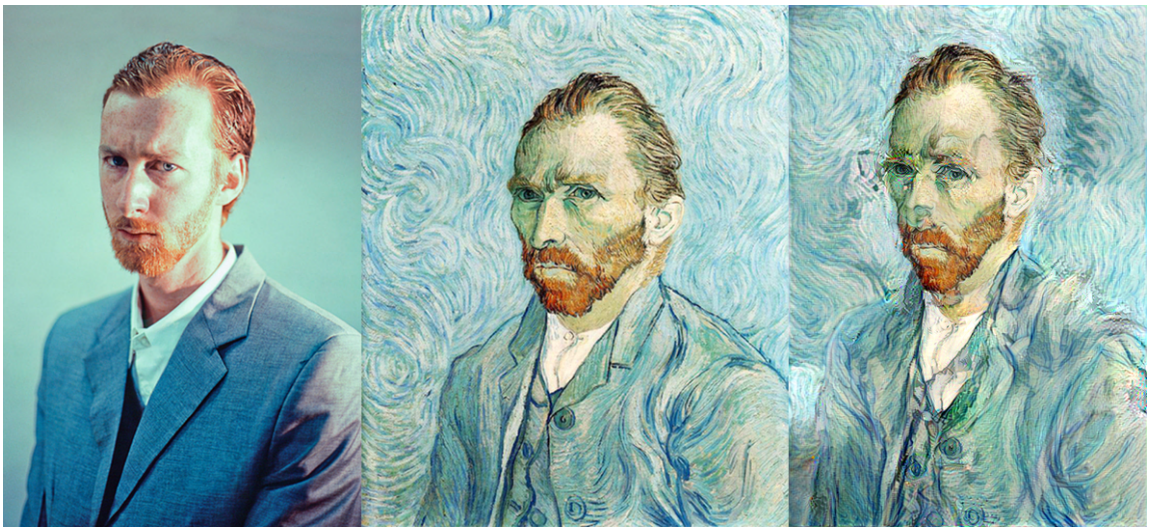


Figure 4: Spatially controlled SST using the method introduced in this work (ODST + Gatys 2017 [13]). Style is transferred between similar regions in the content image (left) [2] and style image (middle) [72], for example in the background, and between facial features (right).

Spatial control in neural style transfer refers to transferring style locally between corresponding regions of the content and style images. Such corresponding regions can be represented by semantic segmentations. Spatial control allows different parts of a style transferred image to be rendered in different local styles present in a style image. This is particularly important in the context of preserving style features, as without the appropriate local style, such features will not be apparent in a rendered style-transfer output.

Gatys *et al.* (2017) [13] outline two methods of integrating spatial control into the style loss calculation: guided gram matrices and guided sums (a neural-patches approach [31]). The trade-off between these two methods is one of texture quality and computational complexity. This work favours texture quality over computational efficiency as it is integral for representing local features, so the guided gram matrices approach is preferred. However, as a result there are still efficiency complications which must be considered further down the line (section 3.2.2).

Guided gram matrices work by adding guidance channels into the style loss calculation. These channels describe which regions in the guide image should be correlated to which in the style image, and are represented as R pairs of binary masks for each content-style region correspondence (i.e. two corresponding semantic segmentations).

The style loss calculation itself is modified to incorporate the guidance channels into the spatial correlation step, which was previously globally fixed. Regular gram matrices are still used for the loss calculation, but the style features are masked for each in mask $M \in R$ in order to sum optimisation targets for corresponding masked regions:

$$\mathcal{L}_S = \frac{1}{N_S} \sum_{r}^{R} \sum_{\ell}^{N_S} \left(\mathbf{G}\{\mathbf{M}_\ell^r \cdot \mathbf{F}_{S_\ell}(\hat{\mathbf{x}})\} - \mathbf{G}\{\mathbf{M}_\ell^r \cdot \mathbf{F}_{S_\ell}(\mathbf{x}_S)\} \right)^2 \tag{6}$$

It should be noted that in order to perform this masking at each feature layer, the masks must be downscaled in the same fashion that the style image is downscaled as its features are extracted from the feature extraction network (VGG). The structure of the network can be copied (i.e. just pooling layers as to not extract features from the mask) to perform the downscale easily.

The factor of R in this calculation increases the computational complexity significantly, particularly so when dealing with high-resolution images as the scaling of the complexity on the images is polynomial in itself in relation to image scale. In order to run this algorithm on consumer hardware, a limited number of guidance channels must be used to reduce R.

2.2.2 Geometric Style via Warping

Figure 5: Using the work of Liu *et al.* (2021) [39], non-parametric warping can be applied to a content image (left) [39], to apply the geometric style of a style exemplar (middle) [69], and produce a geometric style transfer result (right).

Liu *et al.* (2021) [39] propose a technique for incorporating geometric style within style transfer. Geometric style refers to the shape of depicted features in artwork, which can be extremely different from a feature's realistic counterpart when the artwork is geometrically expressive. The inclusion of geometric style in the style transfer process makes it a more true-to-form art generation method when compared to how humans create art, as most artists do not attempt to exactly copy the form of objects they are interpreting onto the canvas. A conclusion from this work is that texture transfer alone insufficient for expressing certain styles, especially when they are particularly expressive or abstract.

Liu *et al.* train a non-parametric warping network which attempts to warp an arbitrary content image onto the geometry of a style image, regardless of semantic content. This is achieved using a feature correlation process somewhat similar to contextually aware texture transfer methods [52, 34]. In the context of style transfer, a flow map is generated from a content image $\mathbf{x}_C$ which coarsely maps it onto a style image $\mathbf{x}_S$.

The benefit of this technique in relation to preserving style features is that it can provide a coarse mapping $\mathbf{x}_C \mapsto \mathbf{x}_S$. This mapping can act as the basis for building a more fine mapping later on

using other techniques such as semantic segmentation. The generation of this initial mapping is fast and versatile, but also somewhat sensitive with regards to certain depictions in artwork. Optimal results are observed when the semantic content of images roughly align with one another. Even when the same subject is depicted, the network can be quite sensitive to the content, and may have trouble finding a sensible warp.

2.3 Segmenting Artwork Images

Figure 6: Segment-Anything [25] segmentation results on varying artistic styles from occidental to oriental. The segmentation is accurate and fine enough to automatically pick out key objects in each piece. Top row: Source style [64, 70, 75, 67] Bottom row: corresponding segmentation.

Up until recently, the task of segmenting artwork has been a challenging one. The large variance in object texture and geometry from artistic depiction makes distinct visual object classes difficult to accurately distinguish from one another. Out of the box segmentation techniques [56, 51] tend to have poor performance due the domain of artwork images being underrepresented in their training datasets [35, 6, 15]. Extracting meaningful features from artwork images is challenging on its own, and other research aims bridge the domain gap between stylised domains [41], but it does not operate directly within the topic of segmentation. Artwork-specific segmentation methods exist, but lack generality [57] or lack the ability to segment finer features [4].

Segment anything [25] is a large, transformer-based [50] segmentation model trained on a novel large dataset, SA-1B, which contains 400× more ground truth segmentation masks than the next largest alternative [28]. It is perhaps unsurprising that, as a result of the sheer scale of this model,

it offers generally good unprompted zero-shot segmentation results across illustrative artwork and real-life images, making it suitable for enabling the artwork feature correspondence step in the pipeline proposed in this work (section 3.2). The model exhibits good performance on a range of artistic styles from differing geographical regions, for example from occidental to oriental, shown in Figure 6. This is perhaps a result of the geographical origin of training images being well-varied and numerous.

2.4 Facial Landmarking

Facial landmarking refers to the process of locating specific points on the human face, such as the mouth, nose, and eyes. A digital representation of the human face can be built by locating a number of these facial landmarks in a given image. Building such a representation proves useful in a wide variety of tasks, from biometric security, facial and emotion recognition, to film, VFX, and virtual reality applications, among others. The fidelity of a landmark model can be all the way from locating one or two landmarks such as irises, all the way up to a generating a fully-formed mesh of the human face, with hundreds of landmarks. Additionally, the use of facial landmarks has been explored within artwork and for style transfer [57], allowing for the geometry of expressive styles to be modelled and transferred using SST.

Nowadays, the detection of landmarks is performed with machine learning algorithms. At a basic level, landmark detection can be thought of as a supervised learning problem, where an ML algorithm learns to correctly position landmarks on faces using large training datasets of annotated faces as reference [29, 21, 43, 53]. Though deep models now dominate the research scene, simple classifiers such as linear regressors or support vector machines (SVM) can still perform modestly, especially when used in ensemble models, such as cascaded regression [9, 55].

Convolutional Neural Networks (CNNs) have long been a standard for the kind of task landmark detection presents. CNNs are able to extract spatial patterns and structures from images, including facial features, which enables them to accurately position landmarks given a reasonably sized training dataset [16, 10]. More recently, transformers have begun being leveraged to further increase accuracy [32]. Aside from a straight-forward point positioning approach, other methods are able to model the geometry of the face in three dimensions. For example, by fitting a facial mesh to an image, landmarks can be aligned to a face in three dimensions, opposed to two [23].

Figure 7: Facial meshes fitted to various faces via [23], with key facial landmarks (eyes, irises and eyebrows) highlighted.

2.5 Stroke-Based Rendering

Stroke-based rendering is a technique within NPR which aims to produce stylised images which resemble hand-drawn or painterly styles. This is achieved through emulation of brush strokes, mimicking traditional art media such as pencils, pens, paint and watercolour. This is beneficial in the context of mimicking style, as certain artistic styles lend themselves much more naturally to a stroke-based approach, and are difficult to mimic on a pixel-by-pixel basis. Cross-hatching, stippling, and ink wash are notable examples of artistic styles which are difficult to capture without brush stroke simulation.

Stroke-based renders are generated by progressively adding strokes onto a virtual canvas (i.e. blank image). The properties (or 'parameters') of simulated brush strokes are changed depending on the content of the reference image in the region the stroke is placed. This can include orientation, size, curvature, among others. In recent works, deep networks are used to automatically determine the placement of brushstrokes, as well as their properties, dubbed "neural painting" [37, 60, 46]. For example, brush strokes can be placed depending on the curvature of colour or saliency of a specific image region. Stroke-based rendering has also been applied to neural style transfer [26], using both content and a reference style to inform brush stroke placement and its parameters.

Though brush stroke rendering works very well for approximate and loose artistic styles, as well as imitating traditional art media, it should still be considered whether brush stroke rendering is suitable for a given artistic style. Artistic styles which have extremely fine rendering may actually look worse once stroke-based rendering is applied. In these cases, the intention of the artist was to make brush strokes invisible to the human eye, so intentional rendering of these strokes could make the output look strange.

2.6 Technology

This work makes use of deep convolutional neural networks (CNNs) which are associated with a relatively high computational cost. The specific networks used will be outlined later, but generally require no more than open-source ML libraries (such as TensorFlow 2) to get working, and in some places, proprietary software (MATLAB).

The memory usage of techniques in this paper can become high when working with high-resolution images, so there are some hardware performance considerations. Processing time and memory consumption are considered throughout the work to ensure that the proposed solution can run on consumer hardware. Usage of high-performance hardware would simply allow for the generation higher resolution results in a smaller time frame. It is necessary however that there is sufficient hardware for hardware acceleration, i.e. at least a consumer-grade GPU, as the computation of a deep CNN inference and optimisation is too slow on a CPU to be feasible.

All results shown in this paper were generated on mid-range consumer-grade hardware: Intel i5-6600k @ 4.5GHz, NVIDIA GTX 1070 GPU, 16GB RAM.

2.7 Data

The main data sources used in this project are for the provision of content and style images for experimental testing and the generation of results. The deep networks used in this work are pre-trained, so high volumes of training data are not required.

Artwork images are pulled from a manner of different sources, as well as some content images. Specific sources can be found in the latter half of this paper's references. All photographic headshots

are sampled from the CelebAMask-HQ dataset [30], a large dataset containing 30000 headshots of various celebrities, so are not individually cited.

3 Automatic ODST - A General Method

3.1 Overview

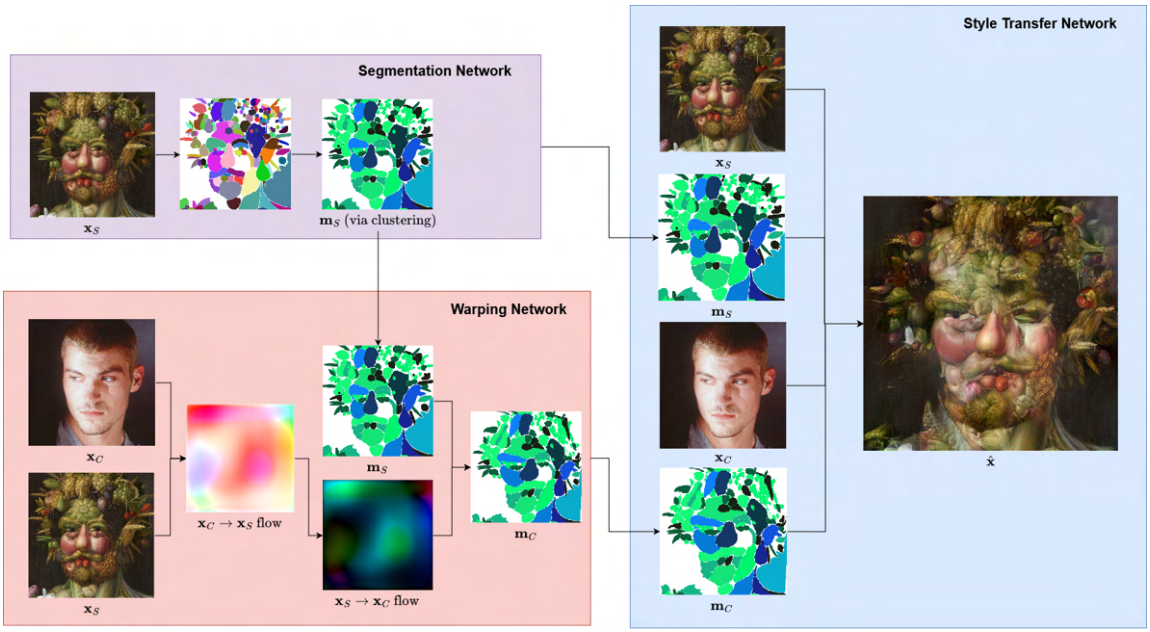
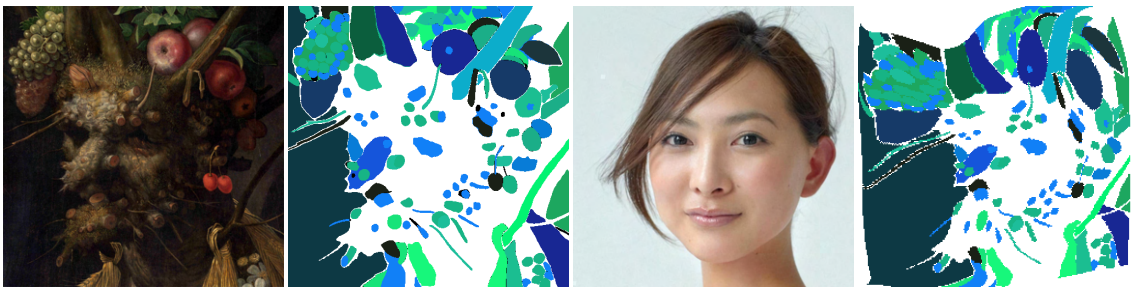


Figure 8: ODST pipeline.

The technique introduced in this work is named Object Driven Style Transfer (ODST). The objective of the technique is to automatically preserve the local features and style of a given style image during style transfer, and to allow its composition to be mimicked in the character of a given content image. The proposed pipeline (Figure 8) enables the autonomous use of segmentation-guided semantic style transfer methods (such as [13, 2, 52]), i.e. usage without having pre-generated manual segmentations of content and style images. Supplementary techniques are explored in the pursuit of computational efficiency and render quality.

3.2 Zero-shot Object Correspondences



$\mathbf{x}_S$ $\mathbf{m}_S$ $\mathbf{x}_C$ $\mathbf{m}_C$

Figure 9: An exemplar content-style image pair ($\mathbf{x}_S$, $\mathbf{x}_C$) and their object correspondence masks ($\mathbf{m}_S$, $\mathbf{m}_C$).

In order to perform the desired spatially controlled SST on a content-style image pair, corresponding segmentations of each image must be generated. To be a truly autonomous and general technique, the corresponding segmentations must be able to be generated on novel data without having to provide any training examples (zero-shot). This means that, under the assumption that the two images are semantically aligned, each feature in the style image should be masked, and have a corresponding mask which roughly matches to a similar feature in the content image. However by the very definition of the problem (preserving local style features), it is not necessarily the case that a feature in the style image will also be present in the content image. Therefore, straightforward segmentation of both content and style images is not applicable. Instead, features of the style image can be segmented, then mapped onto the content image sensibly, creating an 'object correspondence'. This is achieved with two dedicated networks: a segmentation network (SAM [25]) responsible for generating the style image segmentation $\mathbf{m}_S$, and a warping network (Liu *et al.* (2021) [39]) responsible for the mapping from the style segmentation to content segmentation $\mathbf{m}_S \mapsto \mathbf{m}_C$.

3.2.1 Segmentation and Prompting

As introduced in section 2.3, SAM [25] is used to automatically produce a semi-fine segmentation of style images.

Segmentation is never a seamless process, however. It is subjective whether or not a generated segmentation is oversegmented or undersegmented, and there are inevitably cases where certain segments should have been present, or absent. Though prominent (i.e. large, or visually distinct) features are generally properly segmented, it is not uncommon for the model to automatically apply masks over a broader area of distinct features, for example faces, in Figure 10.

Style | Automatic Segmentation (No Prompts) | Adjusted Segmentation (3 Prompt Points) | Manual Segmentation (from [2])

Figure 10: Improving Segment-Anything [25] segmentation results by providing a minimal amount of prompt points about key features (eyes, facial hair). Ground truth segmentation included for reference.

Fortunately, SAM is a promptable model, so the option to manually prompt the model is available in the case that certain key features are missed. Prompts are in the form of positional data (x,y points) and a judgement (positive or negative). Positive points allow the model to attempt to add areas to the mask, and negative points allow for removal. A quick prompting interface implementation allows a user to review an automatic segmentation if desired, and add additional masks or remove them where necessary. This is still orders of magnitude faster than manually creating masks, and typically the model only requires one or two prompt points to segment an individual feature when missed.

3.2.2 Clustering Segments with Affinity Propagation

Style Reference (unsegmented) | Unclustered Segmentation (158 segments) | Clustered Segmentation (19 segments)

Figure 11: The effect of clustering on a segmentation mask. Segment area and corresponding style histogram information were used as the clustering feature vectors.

After an initial style segmentation is produced, there is a need to cluster individual segments together. There are two main motivations for doing so:

1. To reduce the computational complexity of the SST, notably the space complexity.
2. To encourage creative textural rearrangement during the SST.

The main motivation for clustering style segments is for performance - to reduce the memory usage of particular style transfer methods. For a style transfer method like guided gram matrices (section 2.2.1), the number of guidance channels that can be used is limited almost solely by available program memory. Additionally, the more guidance channels that are used, the lower the maximum resolution of the output image. In order to run such a SST method on consumer hardware (in this paper, a GTX1070 GPU and 16GB RAM), and at a reasonable resolution (1MP image as a target), the number of unique guidance channels must be reduced from the number in the initial segmentation. A standard segmentation could produce hundreds of segments, which would make the space complexity explode into infeasibility, whereas a reasonable number of guidance channels to work with is anywhere up to 20 to 30.

Clustering happens to also produce a desirable side effect: greater compositional freedom with regards to texture during SST. By grouping similar objects and features together in a single guidance channel, a SST algorithm is free to draw texture from any of these regions in place of another as it deems fit, often influenced by underlying content features. This allows for a greater level of compositional freedom, as the composition can be effectively shuffled without changing its geometrical properties. More varied style transfer outputs are produced as a result, and they tend to be closer to derivative works than a rote copy of style texture. This is particularly noticeable in this paper's results which use Arcimboldo's works as a style image.

The clustering method used in this pipeline is affinity propagation. Affinity propagation is a hierarchical clustering method which mimics social interaction to produce an unspecified number of homogeneous clusters. Hierarchical clustering methods are appropriate here as an assumption should not be made about the number of distinct object categories in order to keep the technique automatic and versatile. A number of features about individual segmentations are used to perform the clustering. Notably **total segment area**, and **colour histogram information** in the area of the segment masks in the style image. These features alone are enough to segment objects into reasonable categories, in fact, even just area alone. The clusters do not have to be overly distinct and homogeneous, as somewhat mixed clusters will simply enable more compositional freedom for aforementioned reasons.

3.2.3 Correspondence via Geometric Warping

Once a segmentation of the style image is produced ($\mathbf{m}_S$), a mapping $\mathbf{m}_S \mapsto \mathbf{m}_C$ must be generated to complete the object correspondence. It is important to use such a mapping instead of, say, generating a separate segmentation of the content image in order for the segments to correspond to one another across $\mathbf{m}_S$ and $\mathbf{m}_C$. This way, the local features of the style image are preserved in $\mathbf{m}_C$.

The technique introduced by Liu *et al.* (2021) [39] (section 2.2.2) implements a warping network which can be used to produce a coarse mapping from $\mathbf{m}_C$ to $\mathbf{m}_S$ in the form of a flow map. It should be noted that this module is best used to generate $\mathbf{m}_C \mapsto \mathbf{m}_S$ not $\mathbf{m}_S \mapsto \mathbf{m}_C$, as from experimentation, it is easier to warp from a realistic image (content) onto a non-realistic one (style).

Directly warping a complex style example onto a less-detailed content image can produce less than ideal results. Therefore, the warp $\mathbf{m}_S \mapsto \mathbf{m}_C$ should be calculated by simply inverting the generated warp $\mathbf{m}_C \mapsto \mathbf{m}_S$. When applied, this allows the detailed style segmentation $\mathbf{m}_S$ to be fitted to the geometry of $\mathbf{m}_C$, producing the desired object correspondence.

3.3 Spatial Control

Although there are newer techniques in recent research, the primary technique used in this work for spatially-controlled SST is the technique introduced by Gatys *et al.* (2017) [13] (See section 2.2.1 for technical details). This is partly due to its simplicity, allowing it to be easily reimplemented and modified, and also due to the over-aggressive nature of context-aware texture transfer methods. Texture transfer methods being context-aware makes generated images very similar to the source style, for better or for worse. The effect of these methods is discussed later in sections 5 and 6. When content and source images are semantically aligned, and informative warped masks are provided, the resultant generations from context-aware methods almost look like warps of the style image in many cases - too similar to register as a style transfer, or a derivative work of the style image. For these reasons, Gaty's approach will be the main source of experimentation, but a recent context-aware and segmentation-based approach by Wang *et al.* (2022) [52] will also be included for comparison.

It should be noted that this technique does not try to match the geometry of the content image to the style image like other specialised techniques [39, 41]. Because masks are generated from a warped segmentation and local style is enforced via hard masking, the local geometry of the style image can still be implicitly transferred. It would be undesirable to transfer the global geometry of a content image onto a style image alongside local geometry. If it were to be transferred, the resultant image would be conceptually no different to the style image itself, somewhat defeating the point of the technique.

Hard masking refers to binary masks being used in the guided gram matrix calculation. This suppresses the correlation between unmasked areas to zero in the loss calculation, so no style is allowed to blend between unmasked areas. The alternative is soft masking, where masks can take a real-value $\in [0, 1]$, allowing weighted correlation and thus style blending in 'soft' regions. Therefore, the use of hard masking serves to make local features more distinct from one another in style-transfer results by encouraging the generation of visible geometrical boundaries.

3.4 Multi-Scale Strategy

Figure 12: Multi-scale rendering process from 32×32 to 512×512.

A multi-scale strategy can be employed in similar fashion to other texture transfer implementations [39, 49, 52] to progressively build up coarse to fine texture details. Under a multi-scale strategy, style transfer is performed in stages with increasing image sizes corresponding to octaves of a Laplacian pyramid. At each stage, the output of the previous stage is used as the starting guidance image $\hat{\mathbf{x}}_0$. For the first stage, the content and style images can both be used as initial images, with varying results (more on this later).

The immediate positive effects of employing a multi-scale strategy are: 1) reduction of optimisation artifacts (checker-boarding and banding) and 2) a more faithful recreation of the piece, notably with very coherent texture.

Though the texture is more appealing and has better continuity, the character of the content image is somewhat lost. In Figure 12 for example, the resultant image is appealing, but it is not apparent what the content image might have been just by looking at the style-transferred image in isolation. In a roundabout way, the composition of the style image has been changed to match the content (as intended by this pipeline), but to an extreme degree which is potentially beyond what is desirable.

It is possible to adjust the 'contribution' of the content image in a render generated by a multi-scale strategy without changing the associated style transfer weight α. Firstly, allow the style transfer to start with $\hat{\mathbf{x}}_0$ set to a noisy version of the content image. Secondly, skip the lower levels of the multi-scale process. As a result, the style transfer will begin to impose finer details onto the content image from the offset, and the result is often a more content-inspired composition (see Figures 13 and 14).

Figure 13: Comparison of multi-scale outputs with high style composition adherence (left) and low style composition adherence (right).

$\times 32 \rightarrow \times 512$ $\times 64 \rightarrow \times 512$ $\times 128 \rightarrow \times 512$ $\times 256 \rightarrow \times 512$ $\times 300 \rightarrow \times 600$ $\times 512 \rightarrow \times 600$

Figure 14: Differences in multi-scale rendering outputs when progressively reducing the use of lower levels of the Laplacian pyramid. Subfigures 1 to 4 (left) show how content representation can be increased by reducing lower levels. Subfigures 5 and 6 (right) show that content-style representation can still be balanced just by adding more high-level layers.

Overall, the application of a multi-stage rendering strategy is a balancing act between adherence to the original style and composition, and composition creativity. It is advisable to render with some level of multi-scale strategy (e.g. at least two layers) as, in all cases, texture fidelity is increased and generation artifacts are reduced.

3.5 Conclusions

At this point, by composing existing state-of-the-art methods, a state-of-the-art pipeline has been implemented which has a focus of preserving local style features during style transfer. Experimentation within the object correspondence step has shown the importance of generating loose (i.e. with clustering), yet high-quality semantic segmentations for use in SST (enabled by prompting when necessary). Furthermore, the utilisation of a multi-scale strategy is shown to improve output quality from SST, and is a particularly important factor when balancing style composition adherence under SST using object correspondences. Further experimentation results follow in section 5.

4 ODST - Extension to Arbitrary Arcimboldo Headshots

4.1 Overview

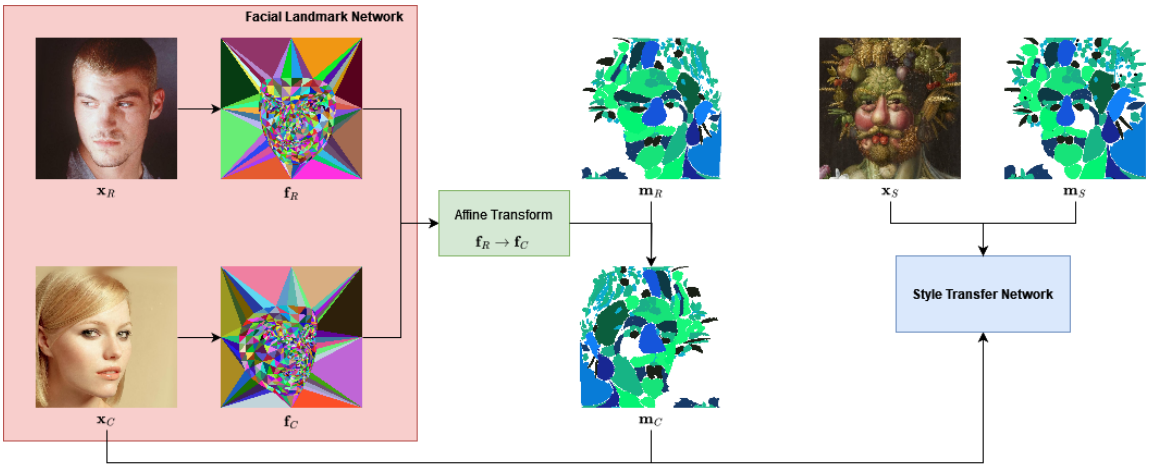


Figure 15: Extension of the ODST pipeline to Arcimboldo headshots.

A significant limitation of the ODST pipeline is within the geometrical warping step (section 2.2.2). A more rigorous evaluation of this limitation can be found in section 7, but in short, the geometrical correspondence step used is quite sensitive to semantic alignment. Though it does not strictly enforce semantic alignment, when objects do not align well, the geometric warping step can fail to find a suitable warp. This can even happen between depicted objects from the same visual object class (VOC). Additionally, this problem is noticeably exacerbated when geometric perspective comes into play - it is not easy to change the perspective of a depicted object via non-parametric warping alone. Warping is generally fine for expressive geometric styles where a depicted object's form is loose and significantly different to reality, but when the geometric style and perspective need to be precise (i.e. for human faces), it can perform suboptimally. In this section, this specific failure case is addressed.

Arcimboldo's works are a high-level benchmark for the performance of ODST, and can particularly suffer from this limitation. When performing SST on Arcimboldo portraits using the ODST pipeline, a face with similar perspective to the target work (style image) *must* be used or else warping will most likely fail. A consequence of this is that any content images with the wrong face alignment will not be able to be properly transferred to. This can be attributed to the fact that the geometric shape of a face changes significantly when the perspective is changed, thus the warping step is operating on partial geometric information only. For example, in an extreme case it is possible to lose over half of the geometrical features of a face when the perspective is changed (i.e. changing from a frontal-face view to side profile).

A solution to this problem is proposed in this section. Under the assumption that the semantic content of content images is constant, the geometry of depicted objects (in this case, realistic faces) can be explicitly modelled in three dimensions, allowing a better object correspondence to be generated for misaligned content images; one which incorporates perspective. The core idea is to

almost turn style transfer on Arcimboldo headshots into a face-filter problem. If it is possible to find one 'training' or 'reference' example $\mathbf{x}_R$ which produces a good object correspondence $(\mathbf{m}_R, \mathbf{m}_S)$ using geometric warping [39] (as in the regular ODST pipeline), the object correspondence can be applied onto other faces by projecting it onto a face mesh of $\mathbf{x}_R$, and transforming onto a new target face mesh of an arbitrary image $\mathbf{x}_C$. We then have the means to generate a new, more accurate object correspondence for faces with arbitrary perspective, which can be used to perform a SST in the same manner as in the general ODST pipeline.

4.2 Building Meshes with Facial Landmarks

Figure 16: Facial landmark detection visualised on a sample of headshot images.

Facial landmark models, as introduced in section 2.4, are models which capture the geometry of human faces via detection and location of a number of key points in a specified region. In this solution, a 468-landmark model is used which well-approximates the geometry of a realistic face, while running quickly [23].

Once the facial landmarks have been detected and positioned, they must be triangulated in order to build up a face-mesh which forms the basis for performing a perspective transform. Procedural methods can be employed such as Delaunay Triangulation [7], or manual creation of edge-maps via 3D modelling also suffices. The method used in this solution is the latter - because facial landmarks are detected using a *canonical* face model (i.e. faces have the same general form for each detection instance), it is easy to manually generate a general mesh edge-map, and make changes as needed later.

One change that has to be made to the canonical face mesh is the inclusion of connections to the image boundaries. These connections allow the mesh to span the entire image, not just the region identified via facial landmarks. The motivation for this is that when a perspective transform is performed from one face to another, the whole image should be transformed into the new perspective, not just the region within the face mesh. Arcimboldo's works are clearly more than just composition in the facial region, so it is important to capture and transform features outside of it too. Thus, there is a need for additional triangles which connect to the image border, which should additionally capture some perspective information about the mesh in question as well. By connecting the borders to landmarks on the mesh silhouette, regions between the face edge and borders will be stretched or compressed when transformed, somewhat aligning features outside the facial region to the transformed face. Using a total of 8 border points is sufficient for this - they are connected to the closest facial regions along the face silhouette (forehead, cheeks, and jaw / chin), and visualised in Figure 17.

Figure 17: Reference faces (top row) and their corresponding facial meshes, generated after landmark detection (bottom row).

An alternative and potentially more accurate approach would be to determine a vertical z-plane which is parallel to the back of the facial mesh and transform this separately, but the outlined approach is preferred in this paper due to its reasonable accuracy given its simplicity.

4.3 Changing Perspective via Piecewise Affine Transforms

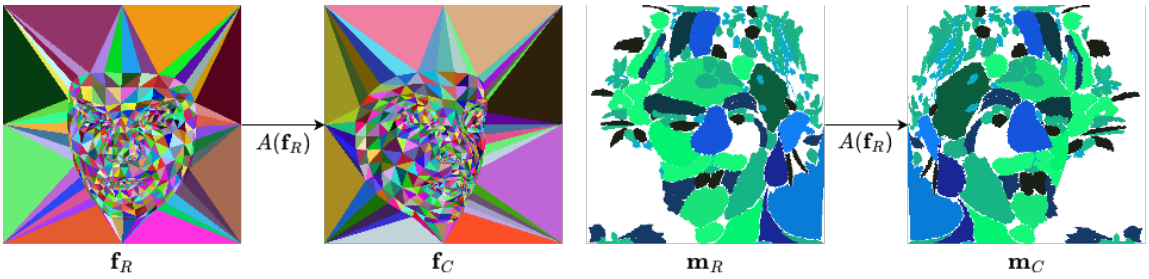


Figure 18: The process of generating a perspective-warped object correspondence using affine transforms. The transform $A(\mathbf{f}_R)$ from $\mathbf{f}_R \rightarrow \mathbf{f}_C$ is reused on the reference segmentation $\mathbf{m}_R$ to produce the desired object correspondence $\mathbf{m}_C$.

Once meshes have been generated for a reference image and an arbitrary content image, the final step is to transform the reference image onto the content image. The motivation is to produce an object correspondence, so the object correspondence of the reference image is transformed in-place of the actual image, though still using the underlying face mesh generated from it.

Since the form of the meshes generated is the same, each triangle in one mesh can be transformed onto the corresponding triangle from the other. This process is called a 'piecewise affine transformation' [5], and is commonly used in filtering, a 'face-morph' being a quintessential example. The

process is not too dissimilar from a regular affine transformation, except that it is performed over sub-regions (triangles), and the region of interest is masked out and drawn to a buffer after it is transformed. Triangles which are closest to the back of the face mesh are drawn to the buffer first to ensure correct layering - a sort of makeshift painter's algorithm.

Figure 18 shows the result of using piecewise affine transformations to change the perspective of a given object correspondence. In some ways, an object correspondence produced this way is actually better than one produced straight from a style image, as the transformation better incorporates perspective by leveraging the 3d positional data given by face meshes.

4.4 Face Orientation and Flipping

Though affine transforms can approximately transform style features onto a new facial mesh ($\mathbf{f}_R \rightarrow \mathbf{f}_C$), performing a full 3-dimensional perspective shift is still not possible. Rendering a style feature from an entirely new viewing angle would be challenging as there is only a single 2d illustration of the feature to work with. The resulting perspective inaccuracy is especially noticeable when a content image and style image have opposite face orientations. Therefore, it is useful to make some observations about the perspective transforms performed and add some heuristic algorithmic improvements to increase output quality.

Perspective shifts can be limited to two cases: left facing and right facing. The perspective of a left-facing object will look roughly correct in any left-facing face, enough so to not immediately appear erroneous. Moreover, performing a left to right perspective shift is extremely simple and only requires flipping the feature about the y-axis. This assumption can be extended to all features in the content, reference, and style images. Therefore, as long as all of the images are facing the same direction (left / right), transferred features will look correctly illustrated in perspective.

To ensure all images face the same direction, a comparison can be made between the content image $\mathbf{x}_C$ and the reference image $\mathbf{x}_R$. The reference image is chosen to be representative of the style image $\mathbf{x}_S$, so they will share the same direction. To ensure $\mathbf{x}_C$ and $\mathbf{x}_R$ share the same direction, the y-axis orientation of their faces meshes $\mathbf{f}_R$ and $\mathbf{f}_C$ can be compared. The calculation of these orientations is a well known procedure, and has an analytic solution formulated by the *perspective-n-point* (PnP) problem [11] when both 3D world points of a mesh and respective 2D camera projection points are available (Equation 7, solved for R).

$$sp_c = K[R|T]p_w \tag{7}$$

where p_c and p_w are corresponding camera and world points, K is the intrinsic property matrix of the virtual camera which visualises the face mesh, and $[R|T]$ is a rotation matrix R of the mesh extended by its translation vector T, i.e.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 & t_1 \\ r21 & r22 & r23 & t_2 \\ r31 & r32 & r33 & t_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{8}$$

Upon solving this equation for R, it is trivial to extract Euler angles for the orientations of $\mathbf{f}_R$ and $\mathbf{f}_C$ in the y-axis. If they are opposite (i.e. their signs do not match), $\mathbf{x}_R$, $\mathbf{m}_R$, $\mathbf{x}_S$ and $\mathbf{m}_S$ are horizontally flipped and $\mathbf{f}_R$ is recalculated before transforming onto $\mathbf{f}_C$.

5 Results

5.1 Automatic ODST on Arcimboldo Headshots

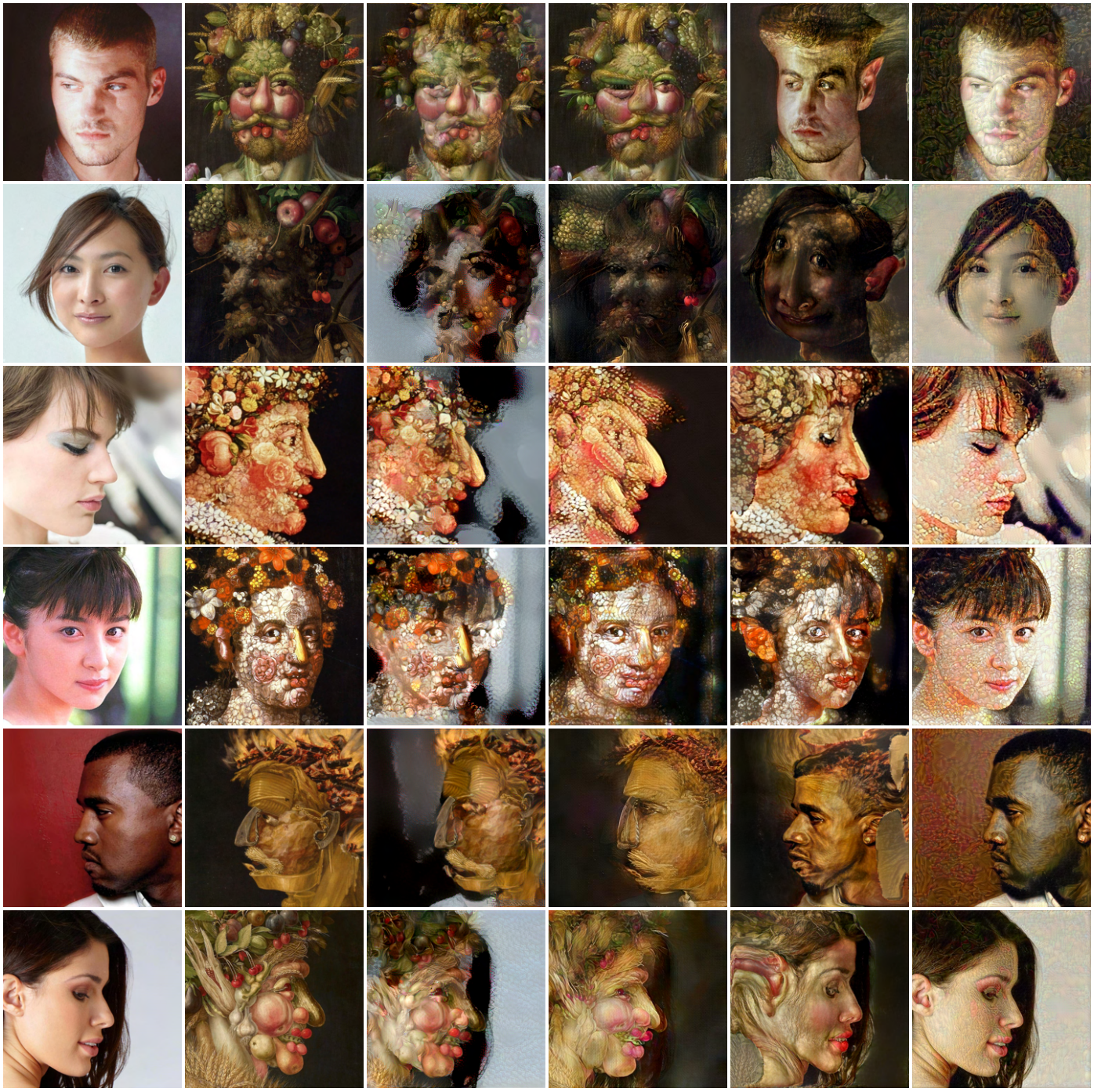


Figure 19: Automatic ODST results on Arcimboldo [65, 62] when used in conjunction with a) spatial control [13], b) context-aware texture transfer [52], compared to geometric style transfer [39] and the original texture-only NST paradigm [12].

5.2 Automatic ODST on Non-Portrait Images

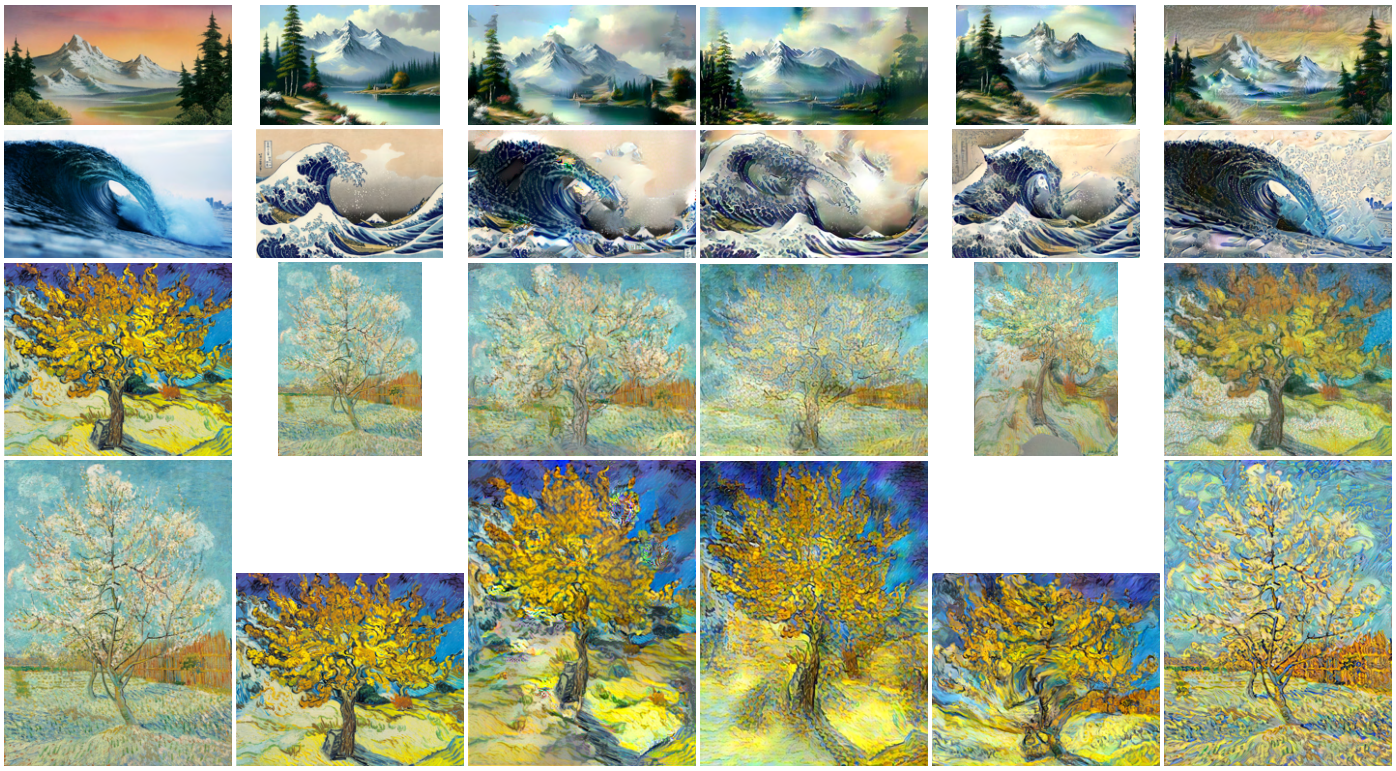
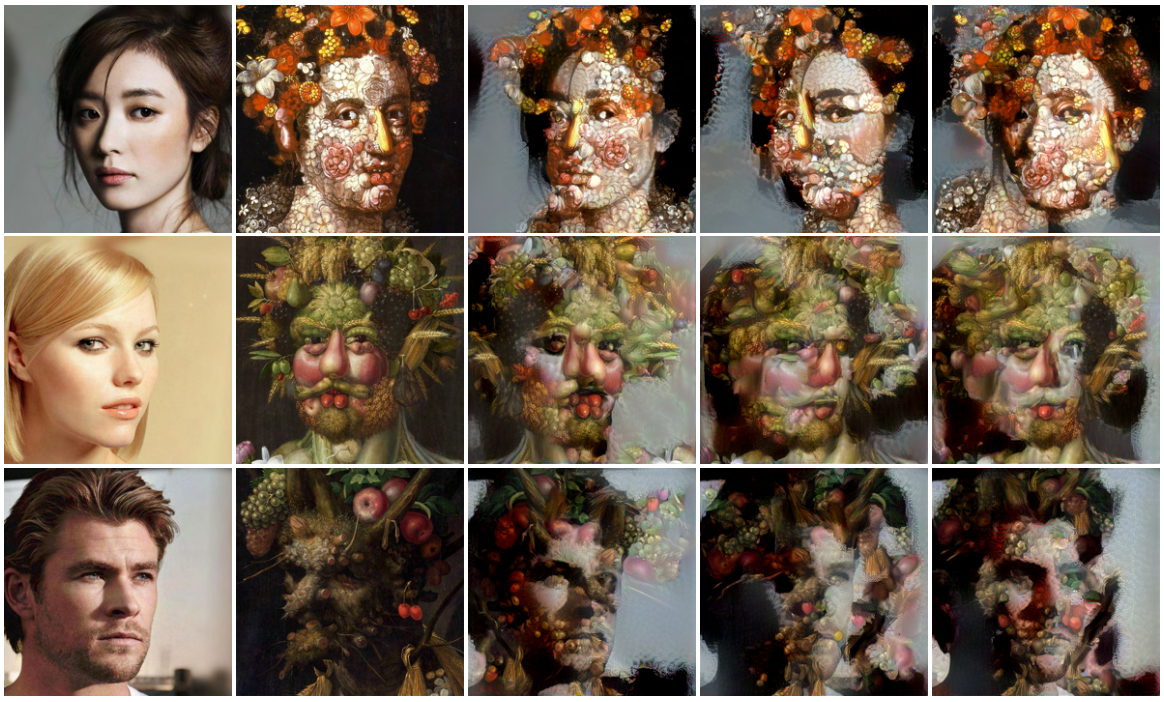


Figure 20: Automatic ODST results on various non-portrait images [73, 71, 74]. Figure format is the same as in Figure 19.

5.3 ODST Extended with Facial Modelling on Arcimboldo Headshots



Content | Style | ODST (Transforms & Flipping) | ODST (Transforms Only) | ODST (General Method)

Figure 21: Results of ODST with facial modelling on Arcimboldo examples when the orientation of the content face mesh $\mathbf{f}_C$ is not aligned with the style image. Usage of both facial transforms and perspective flipping (column 3) is compared to facial transforms only (column 4), and fully automatic ODST (column 5), with Gatys2017 [13] used as the SST method.

6 Discussion

Overall, it is evident that the technique proposed in this work is more suitable for local feature preservation than other recent methods within the field of Style Transfer. Other more sophisticated works fail to perform the task properly despite performing well on other tasks, as local feature preservation is not the focus of their implementation. For example, this can be seen by comparing columns 3 to columns 5 and 6 within figures 19 and 20. This reflects upon how the interpretation of image semantics can majorly effect the outcome of a transfer.

Figures 19 and 20 compare the performance of automatic ODST to similar techniques on both Arcimboldo's works and various artworks with non-human subjects. ODST is able to produce an aesthetically pleasing image which retains the composition of a style exemplar, but in the spirit of a given content image. The geometry of local style features is preserved well, while adhering to the global geometry of the content image. Most large features are transferred quite coherently, and smaller features are still captured, albeit with a little more texture confusion. For example, all Arcimboldo style images in figure 19 have distinct objects to represent the human nose. In all cases using ODST, these objects are coherently transferred to the output image. Floral examples present a challenging case as the majority of features are small (petals, flower buds) and can easily be missed in the segmentation process. However, by allowing the segmentation network to segment smaller areas, or with a little segmentation prompting (see section 3.2.1), this issue is less significant.

When compared to the original work of Gatys *et al.* (2016) [12], it is clear that unguided texture transfer alone is not sufficient for local feature preservation. The geometry of local features is not coherently captured at all in baseline results from their NST method. When compared to the work of Liu *et al.* (2021) [39], we can see how the interpretation of the semantics (geometry) differs between their work and ODST. They are two sides of the same coin in some sense; it could be described that ODST's aim is to fit a style image's geometry to a content image's geometry, and Liu *et al.* aim to do the opposite - fit a content image to a style image. The scales of the geometric transfer also differ. ODST transfers fine geometrical features, whereas Liu *et al.* transfer coarse geometrical features, but overall, ODST is more well suited for this use case.

The SST method that accompanies ODST also significantly affects transfer quality. In figures 19 and 20, the usage of the works of Gatys *et al.* (2017) [13] and Wang *et al.* (2022) [52] is compared. The former enforces strict spatial control, whereas the latter opts for a context-aware approach. Generally, the use of a context-aware approach results in more coherent texture transfer, but local geometry and composition creativity suffers as a result. This is particularly detrimental for most results with Arcimboldo's works as a style exemplar. Though, it can perform similarly or better than strict spatial control on certain examples like floral portraits, as it is generally able to pick up smaller features better. Overall, it appears that the use of a context-aware approach is more suitable on non-portrait images (Figure 20), but performance can still vary within image domains on an image-to-image basis, for better or for worse.

The extension of ODST which includes facial modelling also performs well, allowing generation of Arcimboldo style transfers on arbitrary headshot images. Figure 21 shows the results of this extension applied to content-style pairs where face orientation is mismatched. There are still a few tricky cases (see section 7), but on the whole, transfers are effective. It does not require careful selection of content images to ensure semantics fully align, unlike the general ODST pipeline. Figure 21 also shows that perspective flipping greatly helps produce style transfer results which have consistent perspective when performing geometrical transformations. Overall, the results of this extension show that when the VOC of style transfer images is consistent and can be explicitly

modelled, a reference image can be used to make the transfer algorithm more robust with regards to semantic alignment (i.e. a one-shot algorithm, an algorithm which only requires one training or 'reference' example to produce good results over a variety of novel data).

7 Limitations

7.1 Geometric Warping Sensitivity

The most significant limitation of the pipeline developed in this project is the geometric warping step. The motivation for the extended pipeline with facial modelling (section 4) was as a domain-specific workaround to this limitation. This limitation was touched upon in section 4.1, and it is significant enough to cause issues in a non-negligible number of cases.

The geometric warping step proposed by Liu *et al.* (2021) [39] is sensitive with regards to semantic alignment. Even when two objects from the same VOC are depicted, this step can fail to find a sensible warp from one to the other. Additionally, though the warp is non-parametric (i.e. any pixel can be mapped to any other pixel), it can perform poorly when intricate objects are rotated or re-oriented. A notable example is with faces, with which it is difficult to warp between two faces with different orientations. Figure 21 shows a few examples of resulting failure cases (right-hand column) when the orientation of two faces does not match. Solving the root cause of this issue would be another work in of itself, so instead the limitation can be addressed by implementing domain-specific transformations in place of the warping step in a similar fashion to the extended ODST pipeline outlined in this paper (see section 4.3).

7.2 Generation Artifacts

Figure 22: Various SST generation artifacts: colour banding (left), structure breaks (middle), and colour degradation (right)

Generation artifacts can occasionally appear during SST when using hard spatial control. A few different types were observed during experimentation in this work. It is not always the case that they majorly detract from the transfer outpt, but they are worth considering nonetheless. This may be in part due to some specific implementation quirks (reimplementation of Gatys *et al.* (2017) [13]),

but is also potentially a consequence of the segmentations produced by the segmentation network, and the optimisation method.

Segmentations produced are not completely clean and may have some regions which are small blobs, or outlined by another thin region. Some cleaning steps are used to reduce the occurrence of these regions, but the process is not perfect. During the transfer, these regions can be susceptible to colour degeneration, or significant banding artifacts which may become worse when propagated through levels of a multi-scale output. Similarly, if clustering happens to cluster two very distinct segmentations together (e.g. a dark background and a bright feature), unexpected colour degradation can also occur.

Additionally, if a segmentation is significantly warped, certain features will likely have a significantly different geometrical shape to the source segmentation, and this can cause a patch-mismatch during transfer. The result is a 'break' in the structure of a transferred feature (either by incontinuity, or duplication). Use of a multi-scale strategy helps to reduce these, but it may still occur between significantly warped features.

It is likely that some of these artifacts could be eliminated by changing the optimiser used during the SST. The implementation in this work uses an Adam optimiser [24], which is efficient but may be more prone to generation artifacts. Some implementations make use of more computationally heavy optimisers such as L-BFGS [36] which may produce fewer artifacts. Since the specific SST method is flexible in the ODST pipeline, different implementations can be experimented with as deemed fit to iron out generation artifacts.

7.3 Large Perspective Shifts

Figure 23: A failure case of the extended pipeline when the perspective shift between the reference and given face is too great to transform properly. This is the worst-case scenario, transforming between a side-profile and a frontal-face view.

Though the extended pipeline performs well in a majority of cases, there is at least one major failure case. This is when the perspective shift between the reference face and the given content image is too great. The perspective being overly different means that there is missing information for certain facial regions in the content mesh, which makes it difficult to transfer local information. For example, how can you directly map features onto both halves of a face when only one half is

shown in the reference?

Figure 23 shows the worst case scenario: transforming between a side-profile and frontal-face view. The pipeline replicates features from the style image in sensible positions for half of the face (where the reference has this information). However, one half of the face is left almost empty, only pulling information from the background of the style image as a result of missing information in the reference. Additionally, the positioning of the neck in the reference makes it difficult to transform into the correct position for the content image using the minimal number of triangles that represent non-facial features in this extension (see section 4.2).

Providing a solution to these failure cases is a matter of compromise, but it is likely they could be remedied by a) using the z-plane approach mentioned in 4.2, or additionally modelling neck regions and performing an additional piecewise affine transform. And b) allowing objects which are projected onto facial features to be mirrored on the face mesh when the perspective shift is above a certain threshold (i.e. when the orientations of the content face mesh and reference face differ by, say, 45°).

8 Future Work

There are many avenues for further development of the ideas introduced in this paper. Improvements can first and foremost be made to certain parts of the pipeline to address outlined limitations. Technical improvements could also be made, including, but not limited to:

- Usage of principal component analysis (PCA) during clustering to cluster objects based on not only their area but their shape as well (i.e. round objects vs. long and thin objects).
- Content-aware filling or inpainting of warped segmentations when the image boundary is brought into the image (see $\mathbf{m}_C$ in figure 9). Usually objects are truncated, but they could be extended by a content-aware fill algorithm (for example, PatchMatch [1]).
- A more user-friendly segmentation prompting UI to make the object correspondence step even more streamlined.

Furthermore, ODST could be used alongside alternative style transfer approaches. Though two SST methods were experimented with in this paper, alternatives should be considered to see which produces the best results, especially in the interest of reducing generation artifacts (section 7.2). In somewhat of a different vein, style transfer via parameterized brushstrokes [26] could be explored. This stroke-based approach to style transfer is relatively recent, so enforcing spatial control on it is not an area that has been explored in related literature. If this were to be done, ODST could be applied out-of-the-box, and would likely produce some interesting results as painting by brush strokes is much more true-to-form than optimising pixel colours. This would likely work particularly well for very painterly styles, for example, Van Gogh (see figure Figure 4).

The extended ODST pipeline could also be improved. Neural radiance fields (NeRF) [45] could provide an interesting solution to the perspective shift problem, allowing objects to be properly transformed in 3d perspective along with the face mesh, potentially creating transfer results with an impressive sense of depth. Alternatively, objects in Arcimboldo's works could be modelled by shape (point distribution model [5]) and texture (via texture synthesis, like in [2]). A network could then be trained to place them about certain facial landmarks, creating a work of fusion. Whether or not these ideas are feasible would remain to be seen, but they are interesting food for thought nonetheless.

9 Conclusions

In this work, a semantic style transfer (SST) pipeline has been proposed and implemented, and is able to automatically preserve local features in style exemplars, without the need for manual segmentation. The pipeline is effective on a range of examples, from detailed, surrealist portraits, to landscapes, and images with non-human subjects. An extension pipeline is also introduced which outlines how limitations of the general technique can be overcome within a specific image domain (human faces). Both pipelines are effective and fairly robust, with only a few limitations regarding warp sensitivity, generation artifacts, and large perspective shifts.

The fields of NPR and computer vision are fast moving. Being flexible to new developments is an integral skill when creating novel solutions within them, and the changes made to this work in response to new research (segmentation models, section 2.3) released throughout its timeline reflect that.

Usage of semantic information in style transfer is clearly useful in generating more illustratively accurate works, and is more true-to-form to how humans create artwork. More pleasing transfers can be generated when compared to texture-only style transfer, and particularly in compositionally-heavy artwork, even against more recent SST methods. This work shows that spatial semantics do not have to be manually produced before performing spatially controlled SST. In turn, SST's barrier for entry can be lowered, making it applicable in an increasing number of use cases. The development of style transfer appears to be changing course with the use of semantics, and this contribution has been able to make a subset of those newer paradigms more accessible.

(8807 Words)

References

[1] Connelly Barnes et al. "PatchMatch: A randomized correspondence algorithm for structural image editing". In: *ACM Trans. Graph.* 28.3 (2009), p. 24.

[2] Alex J Champandard. "Semantic style transfer and turning two-bit doodles into fine artworks". In: *arXiv preprint arXiv:1603.01768* (2016).

[3] Ming-Ming Cheng et al. "Structure-preserving neural style transfer". In: *IEEE Transactions on Image Processing* 29 (2019), pp. 909–920.

[4] Nadav Cohen, Yael Newman, and Ariel Shamir. "Semantic Segmentation in Art Paintings". In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, pp. 261–275.

[5] Timothy F Cootes, Cristopher J Taylor, et al. *Statistical models of appearance for computer vision*. 2004.

[6] Marius Cordts et al. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[7] Boris Delaunay et al. "Sur la sphere vide". In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800 (1934), pp. 1–2.

[8] Yingying Deng et al. "Stytr2: Image style transfer with transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11326–11336.

[9] Zhen-Hua Feng et al. "Random cascaded-regression copse for robust facial landmark detection". In: *IEEE Signal Processing Letters* 22.1 (2014), pp. 76–80.

[10] Zhen-Hua Feng et al. "Wing loss for robust facial landmark localisation with convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2235–2245.

[11] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[13] Leon A Gatys et al. "Controlling perceptual factors in neural style transfer". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3985–3993.

[14] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[15] Agrim Gupta, Piotr Dollar, and Ross Girshick. "Lvis: A dataset for large vocabulary instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5356–5364.

[16] Zhenliang He et al. "A fully end-to-end cascaded cnn for facial landmark detection". In: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE. 2017, pp. 200–207.

[17] Hua Huang, Lei Zhang, and Hong-Chao Zhang. "Arcimboldo-like collage using internet images". In: *Proceedings of the 2011 SIGGRAPH Asia Conference*. 2011, pp. 1–8.

[18] Lianghua Huang et al. "Composer: Creative and controllable image synthesis with composable conditions". In: *arXiv preprint arXiv:2302.09778* (2023).

[19] Zixuan Huang, Jinghuai Zhang, and Jing Liao. "Style Mixer: Semantic-aware Multi-Style Transfer Network". In: *Computer Graphics Forum*. Vol. 38. 7. Wiley Online Library. 2019, pp. 469–480.

[20] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[21] Sheng Jin et al. "Whole-body human pose estimation in the wild". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer. 2020, pp. 196–214.

[22] Yongcheng Jing et al. "Neural style transfer: A review". In: *IEEE transactions on visualization and computer graphics* 26.11 (2019), pp. 3365–3385.

[23] Yury Kartynnik et al. "Real-time facial surface geometry from monocular video on mobile GPUs". In: *arXiv preprint arXiv:1907.06724* (2019).

[24] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[25] Alexander Kirillov et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[26] Dmytro Kotovenko et al. "Rethinking style transfer: From pixels to parameterized brushstrokes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12196–12205.

[27] Lironne Kurzman, David Vazquez, and Issam Laradji. "Class-based styling: Real-time localized style transfer with semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[28] Alina Kuznetsova et al. "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale". In: *International Journal of Computer Vision* 128.7 (2020), pp. 1956–1981.

[29] Vuong Le et al. "Interactive facial feature localization". In: *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*. Springer. 2012, pp. 679–692.

[30] Cheng-Han Lee et al. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[31] Chuan Li and Michael Wand. "Combining markov random fields and convolutional neural networks for image synthesis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2479–2486.

[32] Hui Li et al. "Towards accurate facial landmark detection via cascaded transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4176–4185.

[33] Jing Liao et al. "Visual attribute transfer through deep image analogy". In: *arXiv preprint arXiv:1705.01088* (2017).

[34] Yi-Sheng Liao and Chun-Rong Huang. "Semantic context-aware image style transfer". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 1911–1923.

[35] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[36] Dong C Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* 45.1-3 (1989), pp. 503–528.

[37] Songhua Liu et al. "Paint transformer: Feed forward neural painting with stroke prediction". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6598–6607.

[38] Xiao-Chang Liu, Yong-Liang Yang, and Peter Hall. "Geometric and Textural Augmentation for Domain Gap Reduction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14340–14350.

[39] Xiao-Chang Liu, Yong-Liang Yang, and Peter Hall. "Learning to warp for style transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3702–3711.

[40] Xiao-Chang Liu et al. "Depth-aware neural style transfer". In: *Proceedings of the symposium on non-photorealistic animation and rendering*. 2017, pp. 1–10.

[41] Xiao-Chang Liu et al. "Geometric style transfer". In: *arXiv preprint arXiv:2007.05471* (2020).

[42] Yijun Liu et al. "Image neural style transfer with preserving the salient regions". In: *IEEE Access* 7 (2019), pp. 40027–40037.

[43] Ziwei Liu et al. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.

[44] Zhuoqi Ma et al. "Dual-Affinity Style Embedding Network for Semantic-Aligned Image Style Transfer". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[45] Ben Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[46] Reiichiro Nakano. "Neural painters: A learned differentiable constraint for generating brushstroke paintings". In: *arXiv preprint arXiv:1904.08410* (2019).

[47] Robin Rombach et al. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

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

[49] Xavier Snelgrove. "High-resolution multi-scale neural texture synthesis". In: *SIGGRAPH Asia 2017 Technical Briefs*. 2017, pp. 1–4.

[50] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[51] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022. arXiv: `2207.02696 [cs.CV]`.

[52] Zhizhong Wang et al. "Texture reformer: towards fast and universal interactive texture transfer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 2624–2632.

[53] Wayne Wu et al. "Look at boundary: A boundary-aware face alignment algorithm". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2129–2138.

[54] Xiaolei Wu et al. "Styleformer: Real-time arbitrary style transfer via parametric style composition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14618–14627.

[55] Yue Wu and Qiang Ji. "Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3400–3408.

[56] Yuxin Wu et al. *Detectron2*. `https://github.com/facebookresearch/detectron2`. 2019.

[57] Jordan Yaniv, Yael Newman, and Ariel Shamir. "The face of art: landmark detection and geometric style in portraits". In: *ACM Transactions on graphics (TOG)* 38.4 (2019), pp. 1–15.

[58] Wujian Ye et al. "A comprehensive framework of multiple semantics preservation in neural style transfer". In: *Journal of Visual Communication and Image Representation* 82 (2022), p. 103378.

[59] Hui-Huang Zhao et al. "Automatic semantic style transfer using deep convolutional neural networks and soft masks". In: *The Visual Computer* 36 (2020), pp. 1307–1324.

[60] Zhengxia Zou et al. "Stylized neural painting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15689–15698.

Image References

[61] Giuseppe Arcimboldo. *Fire*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[62] Giuseppe Arcimboldo. *Flora*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[63] Giuseppe Arcimboldo. *Four Seasons in One Head*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[64] Giuseppe Arcimboldo. *Rudolf II of Habsburg as Vertumnus*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[65] Giuseppe Arcimboldo. *Spring*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[66] Giuseppe Arcimboldo. *Summer*. Wikimedia Commons (Online). URL: `https://commons.wikimedia.org/wiki/Giuseppe_Arcimboldo`.

[67] Yun Bing. *Flower Study*. Wikipedia (Online). URL: `https://en.wikipedia.org/wiki/Yun_Bing#/media/File:MET_DP153920.jpg`.

[68] *Chureito Pagoda*. Wikimedia Commons (Online). URL: https://commons.wikimedia.org/wiki/File:12-Chureito-pagoda-and-Mount-Fuji-Japan_%2829677439878%29.jpg.

[69] Salvador Dali. *The Peresistence of Memory*. Wikipedia (Online). URL: https://en.wikipedia.org/wiki/File:The_Persistence_of_Memory.jpg.

[70] Jacques-Louis David. *Napoleon Crossing the Alps*. Wikipedia (Online). URL: https://en.wikipedia.org/wiki/Napoleon_Crossing_the_Alps#/media/File:Napoleon_at_the_Great_St._Bernard_-_Jacques-Louis_David_-_Google_Cultural_Institute.jpg.

[71] Vincent van Gogh. *Blühender Pfirsichbaum*. Wikipedia (Online). URL: https://en.wikipedia.org/wiki/Flowering_Orchards#/media/File:Van_Gogh_-_Bl%C3%BChender_Pfirsichbaum.jpeg.

[72] Vincent van Gogh. *Self Portrait, 1889*. Wikimedia Commons (Online). URL: https://commons.wikimedia.org/wiki/Category:Self-portrait_paintings_by_Vincent_van_Gogh#/media/File:Vincent_van_Gogh_-_Self-Portrait_-_Google_Art_Project.jpg.

[73] Vincent van Gogh. *The Mulberry Tree*. Wikimedia Commons (Online). URL: https://commons.wikimedia.org/wiki/File:The_Mulberry_Tree_by_Vincent_van_Gogh.jpg.

[74] Katsushika Hokusai. *The Great Wave off Kanagawa*. Wikimedia Commons (Online). URL: https://commons.wikimedia.org/wiki/File:The_Great_Wave_off_Kanagawa.jpg.

[75] Tōshūsai Sharaku. *Ōtani Oniji III in the Role of the Servant Edobei*. Wikipedia (Online). URL: https://en.wikipedia.org/wiki/Sharaku#/media/File:Toshusai_Sharaku-_Otani_Oniji,_1794.jpg.