An Archetypal Representation of Artistic Style : Summarizing and manipulating artistic style in an interpretable manner

Daan Wynen

Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Daan WYNEN

Thèse dirigée par **Cordelia Schmid**
directeur de recherche, Université Grenoble Alpes
et codirigée par **Julien Mairal**
chargé de recherche, Université Grenoble Alpes

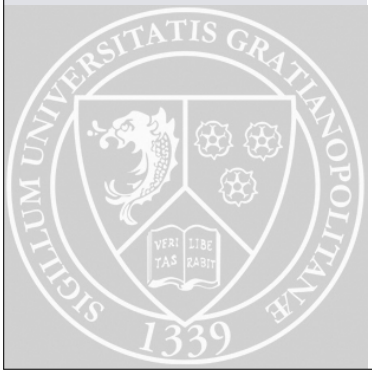
préparée au sein du **Laboratoire Jean Kuntzmann**
et de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

An Archetypal Representation of Artistic Style: Summarizing and manipulating artistic style in an interpretable manner

Une représentation archétypale de style artistique: résumer et manipuler des styles artistiques d'une manière interprétable.

Thèse soutenue publiquement le **9 Décembre 2020**
devant le jury composé de :

Monsieur Patrick Pérez
Directeur scientifique, valeo.ai – Paris, Rapporteur
Monsieur Josef Sivic
Directeur de Recherche, Inria, Équipe WILLOW , Rapporteur
Monsieur Vincent Lepetit
Directeur de Recherche, École des Ponts, Paris, Président
Madame Cordelia Schmid
Directeur de Recherche, Inria, Équipe THOTH, Directeur de thèse
Monsieur Julien Mairal
Chargé de Recherche, Inria, Équipe THOTH, Co-Directeur de thèse



Abstract

In this thesis we study the representations used to describe and manipulate artistic style of visual arts. In the neural style transfer literature and related strains of research, different representations have been proposed, but in recent years the by far dominant representations of artistic style in the computer vision community have been those learned by deep neural networks, trained on natural images. We build on these representations with the dual goal of summarizing the artistic styles present in large collections of digitized artworks, as well as manipulating the styles of images both natural and artistic.

To this end, we propose a concise and intuitive representation based on archetypal analysis, a classic unsupervised learning method with properties that make it especially suitable for the task. We demonstrate how this archetypal representation of style can be used to discover and describe, in an interpretable way, which styles are present in a large collection. This enables the exploration of styles present in a collection from different angles; different ways of visualizing the information allow for different questions to be asked. These can be about a style that was identified across artworks, about the style of a particular artwork, or more broadly about how the styles that were identified relate to one another.

We apply our analysis to a collection of artworks obtained from WikiArt, an online collection effort of visual arts driven by volunteers. This dataset also includes metadata such as artist identies, genre, and style of the artworks. We use this metadata for further analysis of the archetypal style representation along biographic lines of artists and with an eye on the relationships within groups of artists.

Keywords: Archetypal Analysis; Artistic Style; Neural Style; Unsupervised Learning;

Résumé

Dans cette thèse, nous étudions les représentations utilisées pour décrire et manipuler le style artistique d'œuvres d'art. Dans la littérature sur le transfert de style, différentes représentations ont été proposées, mais ces dernières années, les représentations de style artistique qui constituent le paradigme dominant en vision par ordinateur ont été celles apprises par des réseaux de neurones profonds et qui sont entraînés avec des images naturelles. Nous nous appuyons sur ces représentations avec le double objectif de résumer les styles artistiques présents dans de grandes collections d'œuvres d'art numérisées, ainsi que la manipulation des styles d'images naturelles ou artistiques.

Pour cela, nous proposons une représentation concise et intuitive basée sur l'analyse archétypale, une méthode d'apprentissage classique non supervisée avec des propriétés qui la rendent particulièrement adaptée à cette tâche. Nous montrons comment cette représentation archétypale du style peut être utilisée pour découvrir et décrire, de manière interprétable, quels styles sont présents dans une grande collection. Cela permet d'explorer les styles présents dans une collection sous différents angles ; différentes manières de visualiser les résultats d'analyse permettent de poser différentes questions. Ceux-ci peuvent concerner un style qui a été identifié dans la collection des œuvres d'art, sur le style d'une œuvre d'art particulière, ou plus largement sur la relation entre les styles identifiés.

Nous appliquons notre analyse à une collection d'œuvres d'art issues de WikiArt, un effort de collecte en ligne d'arts visuels poursuivi par des bénévoles. Cet ensemble de données comprend également des métadonnées telles que l'identité des artistes, le genre et le style des œuvres d'art. Nous utilisons ces métadonnées pour une analyse plus approfondie de la représentation de style archétypale le long des lignes biographiques des artistes et avec une analyse des relations au sein de groupes d'artistes.

Mots-clefs : Analyse archétypale ; Style artistique ; Style neuronal ; Apprentissage non supervisée ;

Summary in English

The following is a brief overview, chapter by chapter, of the manuscript. It does not include references to sources; for these, please consult the main text. This summary is supposed to convey the gist of the text in a way that is accessible to academics outside of its very specific field or informed laypersons. It should enable the reader to grasp the tasks at hand, the problems involved, the methods of our analysis, and the main outcomes. For readers with substantial background knowledge, it should allow an informed decision on which parts of the main text to select for reading.

Introduction

This work takes a look at representations of artistic style commonly chosen in the computer vision literature. The goal is to build a new, interpretable representation of style on top of these. This representation allows for the analysis of large *collections* of artworks, as well as for the *manipulation* of style of both photos and artworks. We begin by laying out the context of our work. This involves a discussion both of the social function and origin of art, and the technology used to capture, create, and consume art in the 21st centrury.

The last two decades have seen a deluge of digital images being created. Through positive feedback cycles, of both image capture technology and image processing have seen tremendous progress. The development of digital cameras has been especially impressive: they went from specialized scientific instruments and expensive professional photography equipment to commodity hardware that can be bought for a few dollars. Digital cameras are now part of every smartphone, producing millions and millions of images per day.

The capability of digital image processing techniques lagged a few years behind the jump in image capture capability. But since 2012, it has rapidly been catching

up: starting with a record-shattering demonstration of superiority in the task of image classification, the by far dominant form of image processing in the academic computer vision community have been deep neural networks. These loosely biologically inspired methods are able to learn very complex data distributions and have been used to match — and sometimes even surpass — humans at various vision tasks. The tasks that deep neural networks are applied to in the field of computer vision are mostly related to tasks in the real world, and thus involve "natural" images — that is, digital images of natural objects; simply put: photos.

We argue that while the progress that has been — and continues to be — made has been largely focused on photos, the same methods should also be able to improve the treatment of non-natural images, and specifically images depicting artworks. We are by no means the first to suggest this, or to pursue work in this direction. And in fact our work is, at its core, a study of some of the representations that have been chosen by researchers to deal with artistic style. Specifically, we build on the work of Leon Gatys and his collaborators, who in 2015 published a method to use deep neural networks to *modify* photos to look like artworks. This work has been extremely important for the field of artistic style transfer, as this task of image processing is called. But since it is built on a deep neural network representation of artistic style, the method also inherits a key drawback from deep neural networks; namely, that they are utterly incomprehensible to the average human.

We thus set out to study the properties that the neural representations of style have, and to introduce a new representation of style. This representation of style will be derived from the neural representation, but will be built to be (more) interpretable for humans. It will serve us to analyze the content of *collections* of artworks, to be able to ask (and hopefully answer) questions about the relationships of artworks.

Related Work

Before explaining our own work, we lay out how it fits into the history of works that came before it.

The academic work on image processing and art can be traced all the way back to at least the 1960s. At this early stage researchers were interested in how humans perceive texture. Their experiments tested which kinds of textures can and cannot be distinguished by humans. This work on pattern recognition in humans was later picked up by researchers and engineers trying to create *textures*. Creating textures by hand is cumbersome and repetitive, but creating many good textures was an important element of digital art in the context of video games and computer-generated imagery. In the 90s and well into the 00s, the method

of choice for creating textures from a single example were sampling-based; in a first step, the example texture was analyzed to estimate target distributions of features derived from the texture. in the second step, a new texture was sampled according to those distributions. Which distributions to use, which features to extract from the example image, and how to sample the new textures, were the key distinguishing points of the different methods. However, they all lacked the variability that human-made textures showed. They all all exhibited repeating patterns that the human eye can easily single out as unnatural. In light of this, research around the topic slowed down towards the end of the 00s.

In 2015 however, Gatys and his collaborators introduced a new method to generate textures. The defining change they introduced was in the features extracted from the example. While previous methods had all relied on hand-crafted descriptors of style, the method used features learned by a deep neural network. This network had previously been trained on the seeminly unrelated task of classifying photographs. The texture generation method introduced by Gatys and colleagues works by first computing features using this neural network, and then finding, by means of optimization, an image that has similar *statistics* of these features. The statistics they compute are simply the average covariance of the different features computed by the neural network. In line with the experiments from the 1960s, this produces images that show similar visual patterns as the example texture. To generate many new variations of the example, the optimization can simply be started from different, random points.

Previous works had also demonstrated that deep neural networks can be trained to classify artists and styles. In a second work, Gatys and his collaborators thus modified their method to *simultaneously* match feature statistics of one image — which produces images of a *style* matching the example — and the *features* (not their statistics) of another image. This addition allowed them to transfer the style of an artork to a photo, while preserving the content of the photo.

This blew open the field of research into the matter. In the rest of this chapter, we mostly lay out the different improvements and modifications that were introduced to address the various shortcomings of neural style transfer. In summary, it was made faster at the cost of flexibility, and then the flexibility was re-gained step by step, while keeping and even improving the transfer speeds. The latest generation of neural style transfer methods can be executed in realtime on high-resolution video material.

We do go into more detail for one method, namely the “Whitening and Coloring Transform” (WCT) which we use as the basis of our work. It is an elegant method that condenses the entire *transfer* of style into just a handful of mathematical operations, while allowing for arbitrary styles to be transferred onto images, and being relatively light-weight in terms of computation.

One key point of the related works chapter is to demonstrate that despite all the progress that has been made, the representations of style in the neural style literature has not changed. Most of the works use Gatys' chosen representation either directly or indirectly. But as stated before, this representation is not meant for human consumption. It is so high-dimensinal that reasoning about it is out of the question.

We thus proceed to introduce the method we choose to analyze artistic styles in a collection. Archetypal analysis is an unsupervised learning algorithm that was introduced nearly 30 years ago. It allows to analyze data and find common patterns — archetypes — that "explain" the the data it is given. We argue that the way that archetypal analysis relates data points (think: styles of individual artworks) to archetypes (think: groups of styles sharing some characteristics) makes it particularly suitable to our task of exploring art collections.

Unsupervised Learning for Style Analysis and Manipulation

This is the main chapter of the manuscript. It explains how we apply archetypal analysis to a collection of artworks — paintings and drawings, mostly — and shows the results of this analysis.

As mentioned above, Gatys *et al.* choose to represent an artistic style as the covariance of a set of features, extracted by a neural network, and averaged across the locations of the image. They extract these features at different layers of the neural network, resulting in descriptions of the style at different levels of abstraction. To perform style transfer, they take several of these levels of abstraction into account. WCT, which we will use as our our style transfer method of choice, also computes the *mean* feature activation across the different locations of the image.

We thus extract the same features, and concatenate them all into one descriptor of style. As was said before: this representation is somewhat manageable for a computer (it has roughly 200,000 dimensions) but it is certainly not manageable for a human. We reduce the size of this style descriptor by applying a singular value decomposition; essentially, we compress most of the style information into a few dimensions while discarding only very little of the style information. The resulting representation is much smaller (4,096 dimensions) but still nothing a human can reason about. As a second step, we now apply archetypal analysis, choosing the number of archetypes to be adapted to the size of the dataset in question — typically between 32 and 256 archetypes.

The data we use comes from WikiArt; a volunteer-driven project to collect high quality images of artworks, and to annotate them with metadata such as the

artist, when they were painted, and what style (in the art historic sense) they are commonly attributed to.

We collected about 120,000 such paintings from the WikiArt website, and apply the the archetypal analysis to this entire dataset. But we also apply it to subsets of the whole dataset, mainly to demonstrate some properties of the method we propose.

Specifically, we look at datasets of paintings from just a single artist and at datasets of groups of artists that have some art historical relationship. As single artists we investigate the collections of Pablo Picasso, Vincent van Gogh, and Salvador Dalí, some of the most well-represented artists in the WikiArt collection. To demonstrate how our method lets us investigate relationships *between* artists, we analyze the paintings from the Venetian school and a group of four hand-picked notable artists.

The archetypal analysis of styles allows us to summarize the different styles present in the collection being studied. Archetypes typically correspond to a single element of artistic style, such as color or texture. Since they are composed of individual artorks' styles, and in turn explain those styles, this allows different questions to be asked. One of those questions is simply "what kind of styles can we find in this collection?" Another is "how do these styles relate to one another?" Looking at a single artwork, we can ask "which styles can we spot in this artwork?" and "how does it relate to other artworks?"

The different visualization that archetypal style analysis uses to explore these questions all make use of the underlying style *transfer* method. We show how to produce these visualizations, and also show that the archetypes can be used to artistic effect. To achieve this, we slightly modify the underlying WCT transfer method, to allow for better preservation of detail. This is necessary since we modify artworks, not photos, as most style transfer works, but it also adds a new aspect of artistic control that may be useful when applying WCT for artistic purposes.

We finish the chapter with an analysis of the archetypes and their relation to actual art historic concepts. For the analysis itself we do not use the artist and style annotations from WikiArt. However, we are interested to see whether the archetypal representation of style actually manages to capture some of this information, as this would also imply that the underlying style representation capture this information.

When analyzing groups of artists, we do find some hints of archetypes that correspond quite well to certain artists. When analyzing artists that had a clear progression of their style throughout their career, we also see archetypes that capture specific periods. This is most visible in the analysis of Picasso's works. However, all of these connections are quite weak.

We conclude that the underlying representations of style (of which we compared

two) do not sufficiently capture art historic information to unfold their full potential and that of the archetypal analysis.

Conclusion

We conclude the manuscript with a summary of the preceding chapters, and with a bit of discussion of what our analysis actually implies. We surmise that adding supervision to the learning process might help to capture more information of art historic relevance. The data to do so is readily available now; alas, practical issues make this a non-trivial task. The network architectures commonly used for style transfer are quite dated compared to the state of the art in image classification. This makes them hard to train in practice. However, for the task of style transfer, their architecture makes them particularly suitable. Progress on training methods and network architecture may help with these issues however.

Résumé en Français

Voici un bref aperçu, chapitre par chapitre, du manuscrit. Il n'inclut pas de références aux sources primaires; pour ceux-ci, veuillez consulter le texte principal. Ce résumé est censé transmettre l'essentiel du texte d'une manière accessible aux chercheurs en dehors de son domaine très spécifique ou à des amateurs. Il devrait permettre au lecteur de saisir les tâches à accomplir, les problèmes en jeu, les méthodes de notre analyse et les principaux résultats. Pour les lecteurs ayant des connaissances de base substantielles, il devrait permettre une décision sur la sélection des parties du texte principal pour la lecture.

Introduction

Ce travail se penche sur les représentations du style artistique couramment choisi dans la littérature sur la vision par ordinateur. Le but est de construire sur eux une nouvelle représentation du style qui soit plus interprétable. Cette représentation permet l'analyse de grandes collections d'œuvres d'art, ainsi que la manipulation du style des photos et des œuvres d'art. Nous commençons par présenter le contexte de notre travail. Cela implique une discussion à la fois sur la fonction sociale et l'origine de l'art, et sur la technologie utilisée pour capturer, créer et consommer l'art au 21e siècle.

Les deux dernières décennies ont vu un déluge d'images numériques se créer. Grâce à un processus de renforcement mutuel, la technologie de capture d'image et le traitement d'images ont connu d'énormes progrès. Le développement des appareils photo numériques a été particulièrement impressionnant: ils sont passés d'instruments scientifiques spécialisés et de matériel photographique professionnel coûteux à du matériel de base qui peut être acheté pour quelques dollars. Les appareils photo numériques font désormais partie de chaque smartphone, produisant des millions et des millions d'images par jour.

La capacité des techniques de traitement d'images numériques a pris du retard de quelques années par rapport au bond de la capacité de capture d'image. Mais depuis 2012, le traitement d'images a rapidement rattrapé son retard: à partir d'une démonstration de supériorité record dans la tâche de classification d'images, les méthodes dominantes de traitement d'image dans la communauté de la vision par ordinateur a été les réseaux de neurones profonds. Ces méthodes vaguement inspirées biologiquement sont capables d'apprendre des distributions de données très complexes et ont été utilisées pour égaler — et parfois même surpasser — les humains dans diverses tâches de vision. Les tâches auxquelles les réseaux de neurones profonds sont appliqués dans le domaine de la vision par ordinateur sont principalement liées à des tâches du monde réel, et impliquent donc des images «naturelles» — c'est-à-dire des images numériques d'objets naturels; en termes simples: des photos.

Nous soutenons que si les progrès qui ont été réalisés se sont largement concentrés sur les photos, les mêmes méthodes devraient également permettre d'améliorer le traitement des images non naturelles, et plus particulièrement des images représentant des œuvres d'art. Nous ne sommes en aucun cas les premiers à suggérer cela, ni à poursuivre nos travaux dans ce sens. Et en fait, notre travail est, à la base, une étude de certaines des représentations qui ont été choisies par les chercheurs pour traiter du style artistique. Plus précisément, nous nous appuyons sur les travaux de Leon Gatys et de ses collaborateurs, qui ont publié en 2015 une méthode d'utilisation des réseaux de neurones profonds pour modifierphotos pour ressembler à des œuvres d'art. Cet article a été extrêmement important pour le domaine du transfert de style artistique, comme on l'appelle cette tâche de traitement d'image. Mais comme elle est construite sur une représentation de réseau neuronal profond du style artistique, la méthode hérite également d'une inconvéniente caractéristique des réseaux neuronaux profonds; ils sont entièrement incompréhensibles pour l'homme.

Nous avons donc entrepris d'étudier les propriétés des représentations neuronales du style et d'introduire une nouvelle représentation du style. Cette représentation du style sera dérivée de la représentation neuronale, mais sera construite pour être (plus) interprétable pour les humains. Il nous servira à analyser le contenu des collections d'œuvres d'art, pour pouvoir poser (et, espérons-le, répondre) des questions sur les relations des œuvres d'art.

Travaux connexes

Avant d'expliquer notre propre travail, nous exposons comment il s'inscrit dans l'histoire des œuvres qui l'ont précédé.

Les travaux universitaires sur le traitement de l'image et l'art remontent au

moins aux années 1960. À ce stade précoce, les chercheurs s'intéressaient à la façon dont les humains perçoivent la texture. Leurs expériences ont testé quels types de textures peuvent et ne peuvent pas être distingués par les humains. Ce travail sur la reconnaissance de motifs visuels chez l'homme a ensuite été repris par des chercheurs et des ingénieurs essayant de créer des textures numériques. La création de textures à la main est difficile et répétitive, mais la création de nombreuses bonnes textures était un élément important de l'art numérique dans le contexte des jeux vidéo et de l'imagerie générée par ordinateur. Dans les années 90 et jusque dans les années 2000, la méthode de choix pour créer des textures à partir d'un seul exemple était basée sur l'échantillonnage; dans un premier temps, l'exemple de texture a été analysé pour estimer les distributions cibles des caractéristiques dérivées de la texture. Dans la deuxième étape, une nouvelle texture a été échantillonnée selon ces distributions. Les distributions qui étaient utilisées, les characteristics à extraire de l'image d'exemple et la manière d'échantillonner les nouvelles textures étaient les principaux points de distinction des différentes méthodes. Cependant, ils manquaient tous de la variabilité que les textures fabriquées par l'homme montraient. Ils présentaient tous des motifs répétés que l'œil humain peut facilement qualifier de non naturels. À la lumière de cela, les recherches sur le sujet ont ralenti vers la fin des années 2000.

En 2015 cependant, Gatys et ses collaborateurs ont introduit une nouvelle méthode pour générer des textures. Le changement principal qu'ils ont introduit concernait les caractéristiques extraites de l'exemple. Alors que les méthodes précédentes reposaient toutes sur des descripteurs de style fabriqués à la main, la méthode utilisait des caractéristiques apprises par un réseau neuronal profond. Ce réseau avait déjà été formé à la tâche de classification des photographies, à première vu sans rapport au styles artistique. La méthode de génération de texture introduite par Gatys et ses collègues fonctionne en calculant d'abord des caractéristiques à l'aide de ce réseau neuronal, puis en trouvant, au moyen de l'optimisation, une image ayant des statistiques similaires. Les statistiques qu'ils calculent sont simplement la covariance moyenne des différentes caractéristiques calculées par le réseau neuronal. Conformément aux expériences des années 1960, cela produit des images qui montrent des motifs visuels similaires à ceux de l'exemple de texture. Pour générer de nombreuses nouvelles variantes de l'exemple, l'optimisation peut simplement être lancée à partir de différents points aléatoires.

Des travaux antérieurs avaient également démontré que les réseaux de neurones profonds peuvent être formés pour classer les artistes et les styles. Dans un second article, Gatys et ses collaborateurs ont ainsi modifié leur méthode pour faire correspondre simultanément les statistiques de caractéristiques d'une image — qui produit des images d'un style correspondant à l'exemple — et les caractéristiques (et non leurs statistiques) d'une autre image. Cet ajout leur a permis de transférer

le style d'une œuvre d'art sur une photo, tout en préservant le contenu de la photo.

Cela a ouvert le champ de la recherche en la matière. Dans le reste de ce chapitre, nous exposons principalement les différentes améliorations et modifications qui ont été introduites pour remédier aux différentes lacunes du transfert de style neuronal. En résumé, il a été rendu plus rapide au détriment de la flexibilité, puis la flexibilité a été regagnée étape par étape, tout en conservant et même en améliorant les vitesses de transfert. La dernière génération de méthodes de transfert de style neuronal peut être appliquée en temps réel sur du matériel vidéo haute définition.

Nous allons plus en détail sur une méthode, à savoir la «Transformation de blanchiment et de coloration» (WCT : "Whitening and Coloring Transform") que nous utilisons comme base de notre travail. C'est une méthode élégante qui condense tout le transfert de style en une poignée d'opérations mathématiques, tout en permettant le transfert de styles arbitraires sur des images et en étant relativement légère en termes de calcul.

Un point clé de ce chapitre consacré aux travaux connexes est de démontrer que malgré tous les progrès qui ont été réalisés, les représentations du style dans la littérature des styles neuronaux n'ont pas changé. La plupart des travaux utilisent la représentation choisie par Gatys directement ou indirectement. Mais comme indiqué précédemment, cette représentation n'est pas destinée à la consommation humaine. Il est si haut dimensionnel qu'il est hors de question d'en raisonner.

Nous procédons ainsi à introduire la méthode que nous choisissons pour analyser les styles artistiques dans une collection. L'analyse archétypale est un algorithme d'apprentissage non supervisé qui a été introduit il y a près de 30 ans. Il permet d'analyser les données et de trouver des modèles communs — des archétypes — qui «expliquent» les données qui leur sont données. Nous soutenons que la façon dont l'analyse archétypale relie les points de données (pensez: styles d'œuvres d'art individuelles) aux archétypes (pensez: groupes de styles partageant certaines caractéristiques) la rend particulièrement adaptée à notre tâche d'exploration des collections d'art.

Apprentissage non supervisé pour l'analyse et la manipulation de style

C'est le chapitre principal du manuscrit. Il explique comment nous appliquons l'analyse archétypale à une collection d'œuvres d'art — peintures et dessins, principalement — et montre les résultats de cette analyse.

Comme mentionné ci-dessus, Gatys choisit de représenter un style artistique comme la covariance d'un ensemble de caractéristiques, extraites par un réseau de neurones et moyennées à travers les positions dans l'image. Ils extraient

ces caractéristiques à différentes couches du réseau neuronal, aboutissant à des descriptions du style à différents niveaux d'abstraction. Pour effectuer un transfert de style, ils prennent en compte plusieurs de ces niveaux d'abstraction. WCT, que nous utiliserons comme notre méthode de transfert de style de choix, calcule également l' activation moyenne des fonctionnalités à travers les différents positions dans l'image.

Nous extrayons ainsi les mêmes caractéristiques, et les concaténons toutes en un seul descripteur de style. Comme on l'a dit précédemment: cette représentation est quelque peu gérable pour un ordinateur (elle a environ 200000 dimensions) mais elle n'est certainement pas gérable pour un humain. Nous réduisons la taille de ce descripteur de style en appliquant une décomposition de valeur singulière; essentiellement, nous compressons la plupart des informations de style en quelques dimensions tout en supprimant très peu d'informations de style. La représentation résultante est beaucoup plus petite (4 096 dimensions) mais rien sur lequel un humain ne peut raisonner. Dans un deuxième temps, nous appliquons maintenant l'analyse archétypale, en choisissant le nombre d'archétypes à adapter à la taille du jeu de données en question — typiquement entre 32 et 256 archétypes.

Les données que nous utilisons proviennent de WikiArt; un projet mené par des bénévoles pour collecter des images de haute qualité d'œuvres d'art et les annoter avec des métadonnées telles que l'artiste, le moment où elles ont été peintes et le style (au sens historique de l'art) auquel elles sont communément attribuées.

Nous avons collecté environ 120000 de ces peintures sur le site Web de WikiArt et appliquons l'analyse archétypale à l'ensemble de cet ensemble de données. Mais nous l'appliquons également à des sous-ensembles de l'ensemble de données, principalement pour démontrer certaines propriétés de la méthode que nous proposons.

Plus précisément, nous examinons des ensembles de données de peintures d'un seul artiste et des ensembles de données de groupes d'artistes qui ont une relation historique de l'art. En tant qu'artistes, nous étudions les collections de Pablo Picasso, Vincent van Gogh et Salvador Dalí, certains des artistes les plus représentés de la collection WikiArt. Pour démontrer comment notre méthode nous permet d'étudier les relations entre artistes, nous analysons les peintures de l'école vénitienne et d'un groupe de quatre artistes notables triés sur le volet.

L'analyse archétypale des styles permet de résumer les différents styles présents dans la collection étudiée. Les archétypes correspondent généralement à un seul élément du style artistique, comme la couleur ou la texture. Comme ils sont composés de styles d'œuvres d'art individuelles et expliquent à leur tour ces styles, cela permet de poser différentes questions. L'une de ces questions est simplement "quel genre de styles pouvons-nous trouver dans cette collection?" Un autre est "comment ces styles sont-ils liés les uns aux autres?" En regardant une seule œuvre d'art, nous pouvons nous demander "quels styles pouvons-nous repérer dans cette

œuvre?" et "comment est-il lié aux autres œuvres d'art?"

Les différentes visualisations que l'analyse de style archétypale utilise pour explorer ces questions utilisent toutes la méthode de transfert de style sous-jacente. Nous montrons comment produire ces visualisations, et montrons également que les archétypes peuvent être utilisés à des fins artistiques. Pour y parvenir, nous modifions légèrement la méthode de transfert WCT sous-jacente, afin de permettre une meilleure préservation des détails. Cela est nécessaire car nous modifions les œuvres d'art, pas les photos, comme la plupart des travaux de transfert de style, mais cela ajoute également un nouvel aspect de contrôle artistique qui peut être utile lors de l'application de WCT à des fins artistiques.

Nous terminons le chapitre par une analyse des archétypes et de leur relation avec les concepts historiques de l'art. Pour l'analyse elle-même, nous n'utilisons pas les annotations d'artiste et de style de WikiArt. Cependant, nous sommes intéressés de voir si la représentation archétypale du style parvient réellement à capturer certaines de ces informations, car cela impliquerait également que la représentation de style sous-jacente capture ces informations.

En analysant des groupes d'artistes, nous trouvons quelques indices d'archétypes qui correspondent assez bien à certains artistes. En analysant des artistes qui ont eu une progression claire de leur style tout au long de leur carrière, nous voyons également des archétypes qui capturent des périodes spécifiques. Ceci est le plus visible dans l'analyse des œuvres de Picasso. Cependant, toutes ces connexions sont assez faibles.

Nous concluons que les représentations sous-jacentes du style (dont nous avons comparé deux) ne captent pas suffisamment les informations historiques de l'art pour déployer leur plein potentiel et celui de l'analyse archétypale.

Conclusion

Nous concluons le manuscrit par un résumé des chapitres précédents et par un peu de discussion sur ce que notre analyse implique réellement. Nous supposons que l'ajout d'une supervision au processus d'apprentissage pourrait aider à capturer plus d'informations ayant une pertinence historique de l'art. Les données pour ce faire sont désormais facilement disponibles; hélas, des problèmes pratiques en font une tâche non triviale. Les architectures de réseau couramment utilisées pour le transfert de style sont assez anciennes par rapport à l'état de l'art en matière de classification d'images. Cela les rend difficiles à former dans la pratique. Cependant, pour la tâche de transfert de style, leur architecture les rend particulièrement adaptés. Les progrès sur les méthodes de formation et l'architecture du réseau peuvent toutefois aider à résoudre ces problèmes.

Contents

Chapter 1

Introduction

1.1 Context

Visual art has been an important means of expression for mankind for tens of thousands of years. The earliest visual art that we know of, simple hand stencils on cave walls, have been dated to over 64,000 years ago[29]. Visual art, or art in general, offered early humans a means for expression of self, as well as expression of other concepts and interpretations of the world around them. Each year, hundreds of thousands of tourists visit caves to see these expressions for themselves, to understand what moved our ancestors. While less clearly defined than written language, which evolved much later, visual art serves to transmit information, emotions, and interpretations between individuals in a universal and simultaneously very personal way. And with the capability of creating art thus comes the task of interpreting art created by others. It is a task that requires knowledge of the artwork itself, but also the context of its creation, the situation of the artist, and often specific expertise of the techniques employed in the artistic process. To a large degree, human experts perform these tasks intuitively. For non-abstract artworks, the interpretation is often straight-forward and closely related to the processing of natural images, a task that we spend significant amounts of energy and brain activity on.

But a master of their art can also use it to invoke direct associations and emotions in others, simply by use of color, texture and other aspects of the materials in use. Figure 1.1 shows two examples at opposing sides of the spectrum: Willem Claesz. Heda put his mastery of oil and brushes to the end of achieving the most realistic


Figure 1.1 – Left: Willem Claesz. Heda's "Still life with oysters, a rummer, a lemon and a silver bowl" (1634) Right: Wassily Kandinsky's "Composition IV" (1913)

depiction of his still lives; Wassiliy Kandinsky on the other hand used the same materials to create a visual experience completely devoid of realism, but which still "speaks to" to the viewer. This is intentional: Kandinsky indeed wanted to visualize the concepts of flood, baptism, destruction, and rebirth, but chose to do so in an abstract manner. Another bit of context that is critical for the interpretation of a work of art is its relation to other pieces by the same artist, as well as to works of other artists. Since artists often teach, influence, and collaborate with each other, so too their works show similarities that the trained eye can spot and use to contextualize a work. As such, the task of artwork contextualisation falls into the category of visual processing that has seen such tremendous progress in recent years. A lot of work has gone into models that are able to describe the content of images, and it stands to reason that the progress that has been made can be built upon to benefit the domain of visual arts as well.

Especially natural images, meaning images taken by (digital) cameras, have become utterly ubiquitous in the modern world, and have thus received the bulk of the attention of researchers. Through the use of end-to-end differentiable models, previously untractable problems in vision have been solved with the quality of results rivalling, and sometimes surpassing, that provided by human domain experts. What fuels the research in this domain is a combination of availability of natural images and the necessity to sift through them and make them useful. As a result, all manner of old and new problems have seen benefits from the progress of these methods. This leads to a virtuous cycle: more images get taken, because it is more and more affordable to do so, and because photos can be used for more and different purposes, creative or otherwise. This motivates more research in computer vision, since increased availability allows addressing problems for which there was previously not enough data available, and because the large collections that are

newly produced come with their *own* set of problems around curation, exploration, filtering, copyright and other issues. The newly developed methods in turn lead to new applications, at which point the circle repeats with more work on image capture technology and image processing.

While this has been playing out very publicly over the last two decades or so, the most direct impact of the improved methods of computer vision has always been limited to the domain of *natural* images. However, the world of art and art history has been waking up to the possibilities of digital technology for the purposes of cataloging and education as well. Museums all around the world, public and private, have been digitizing their catalogs and have been working on and adopting standards like the Open Archives Initiative Protocol for Metadata Harvesting[42] for making available both metadata as well as images of artworks. Simultaneously, volunteer-driven projects like WikiArt [1] and WikiData [2] are building catalogs of high-quality images taken of artworks around the world, and are annotating them with metadata. On top of these efforts, online platforms such as DeviantArt [3] facilitate the creation and distribution of new, original artworks at an unprecedented rate, democratizing the production and dissemination of artworks in the process. Thus, while the number of artworks publically available is certainly some orders of magnitude smaller than that of natural images, there now exists a sufficient number of them — available to anyone — that navigation and exploration become an issue. On the other hand, these large numbers of images allow the application of exactly those large scale learning techniques that have upended the field of computer vision in recent years.

This development explains both the motivation and the feasibility of applying modern computer vision methods to the field of visual arts. But as nearly every technological advancement before it, the availability of large collections of artworks, and the methods for processing them, also attracts certain artists and enables them to express themselves in novel and unforseen ways. Since the creation of computers, they have been used by artists to create novel forms of art. Early examples of this include Desmond Paul Henry's drawing machines from the 1960s, as well as "COMPUTER PROGRAM FOR ARTISTS: ART I*" by Nash and Williams in 1970 "Computer Program for Artists".

Qickly, digital visual art became more than just a digital version of canvas and paint: as graphics processing, and with it video games, became more advanced, good textures became an important aspect of the visual quality (and thus commercial performance) of video games. Making good textures is hard work though, and repeating the same texture over and over, while technically straight-forward to do,

1. https://wikiart.org
2. https://www.wikidata.org
3. https://www.deviantart.com

is very obvious to the human eye. In the 1990s, engineers and researchers were thus interested in generating new textures as non-repeating variations of old ones, to use in video games and similar applications. The sampling based methods developed during this time were limited by the computational resources available though, and stagnated in the years prior to 2010. Interest in these methods has recently been revived by a series of works focusing on professional animation production techniques.

In recent years however, the by far dominant form of visual art being investigated in an academic an engineering setting is that of neural style transfer. Neural style transfer makes use of methods of deep learning that have rapidly taken over in the computer vision community. These loosely biologically inspired methods are able to learn very complex data distributions and are — sometimes counter-intuitively — applicable to a very wide range of problems. In a series of works Gatys et al. demonstrate that the functions learned by neural networks for the task of image classification can be used to transfer the artistic style of one image to another, while preserving the image content. These techniques have since been improved and adapted for use by enthusiasts and even by consumers, in the form of simple to use mobile applications or websites.

While neural style transfer, and the work building on it, clearly mark a significant development in the field of generative art, at first sight it may not be clear to everyone why they even work. That is not to say that these methods come out of thin air; they are, like the sampling-based methods before them, built on psychological insights reaching as far back as the 1960s. And while the results that neural style transfer produces can be stunning, especially when employed by capable artists as part of a bigger toolkit, there remain failure cases that not only serve as a challenge, but also open up new lines of investigations.

It is in this context that we investigate the influence that the representations chosen for artistic style transfer have, and how these representations can also be used for analytical purposes instead. This investigation will turn out to provide guidance for further work on style *manipulation*. In the next section, we will lay these goals out in more detail.

1.2 Goals and Challenges

This dissertation investigates the representations used in the literature on neural methods of artistic style transfer, using them for the purpose of analysis and to gain an understanding of their strengths and shortcomings.

One important aspect of art history are the relationships *between* artists and how they influence each other's works. To gain insight into these relationships, but also simply for exploring without any particular aim, the study of collections

of artworks can be interesting for experts and laypersons alike. Navigating a big corpus of artworks can be quite overwhelming though, and without significant time spent in training it is easy to miss connections between artworks and styles that would be evident to an expert. At the same time, many artworks' categorizations into styles and genres is the subject of lively discussions at dinner parties and thelike. This is not due to a lack of sophistication on the part of art historians and critics though. The degrees of freedom that artists possess and readily explore make the task of structuring the corpus of *already existing* artworks quite difficult. And in fact, even a good categorization scheme will not stay that way forever since these degrees of freedom are not explored randomly; many artists strive to innovate and re-invent their art with every work, trying to escape any pre-established categories. As a result, the description of style as a simple hierarchy of categories often fails to capture the nuance of the subject matter. Despite all of that however, there exists significant consensus about many groups of artists and artworks being related or similar in some ways. These groups are often defined by shared acquaintances or philosophical underpinnings that don't necessarily manifest as visual resemblance. Many of them do however also show very clear visual commonalities.

Art scholars can often place artworks into the right context without information about when and by who it was created. They can perform this contexturalization task intuitively and based solely on the appearance of the artwork itself, as well its similarities with styles of other artworks they have seen before. This implies that it should be possible to build representations of artwork imagery that capture these relationships too. These representations could be used for exploration of artworks, allowing for similarities and influences between artworks to be spotted and reasoned about. Methods from the modern computer vision toolkit seem like good candidates for approaching this task. These recent methods also allow a sufficiently descriptive representation to be used for image *manipulation*. However, current representations used for style manipulation, while useful for that task, are very high-dimensional and not interpretable to humans, which makes them less useful for the exploration tasks.

Bridging this divide is one of our goals. We propose a method for the analysis of collections of artworks, allowing for connections between artworks to be drawn in an interpretable way. This is achieved by the combination of two properties: First, the representation must be *concise* enough for a human to be able to reason about them. Most representations of style — aimed as they are at machine consumption for the purpose of style manipulation — do not meet this criterion. Second, the representation should be as intuitively *meaningful* as possible. A concise representation does not inherently allow for interpretation; in fact, many methods for representing information in the most concise way possible result in shortened representations that are utterly incomprehensible to the average human.

This is because they typically try to preserve as much information as possible, resulting in descriptions that mix concepts together to increase the information density. However, humans are not able to hold arbitrary numbers of concepts in their mind, and so for a concise description to be meaningful for us, it needs to separate out a few meaningful properties and describe those.

Our method combines archetypal analysis [10], a classic unsupervised machine learning algorithm, with the style representations chosen by several works in the domain of neural style. This approach brings with it a dependence on the underlying representations and the aspects they are able to capture. One goal is thus to assess if these representations are *capable* of capturing information that is useful for the exploration and summarization of artwork collections. We invesigate the influence that the choice of style representation has on the outcome of the analysis, and draw conclusions about the qualities of the representations themselves.

Our method gives an interpretable view on the styles and the artworks in a collection and what their relationships are. The representation we propose is consice enough to be interpretable, and in combination with the underlying style manipulation techniques can be used to explore a dataset by visual inspection. Depending on the situtaion, conciseness and expressivity can be traded off, too. This way, collections can be analyzed for the different styles present in them, and individual artworks can be analyzed in terms of the styles in the collection that they relate to.

Since we are building on neural style representations we inherit their notion of relying on simple real-valued vectors and matrices to represent a style. It would be conceivable to model representations of style in different ways, like hierarchical representations or nearest neighbour classifications. Examples of this include [63] and [46]. However, neural methods make it particularly easy to reuse descriptions for manipulation, which is why we choose to stay within this framework.

Using the archetypal representation of style for an artwork or a collection of artworks, we can then set out to use them for the manipulation of images. These can be natural images, as is the case for most neural style applications, or they can themselves be artworks. Neural style methods can — in principle and in many implementations — interpolate between different styles. Since our analysis builds upon the same representations of style it inherits this capability. This directly results in an intuitive way of controlling the style of arbitrary input images, be they part of the collection or not. Depending on the collection being analyzed and the hyperparameters used for analysis, the summarized styles from the collection can be seen as a pre-selected collection of salient styles which allow a user to pick and choose styles to apply to any input image, resulting in styles that were not necessarily seen in the input collection. While this can be convenient, it is not qualitatively different or necessarily better than interpolation between several,

manually curated, styles from arbitrary artworks. However, the archetypal analysis also gives us the possibility to analyze artworks that were not part of the collection to be analyzed. This is a case not commonly covered by style manipulation methods, since they usually aim to manipulate natural images and not images of artworks. Our method allows us to work with these images and change them in subtle ways. This is achieved by first computing an archetypal representation of their style, and then making changes to that representation. The result is a subtle manipulation of the input's style that mostly preserves its appearance. This can be useful for procedurally generated textures and decorations and similar applications.

As an academic endeavour, the discussion of artworks and their style has a long tradition and is the subject of the field of art history. While single texts on certain aspects of art can be found in writings as early as ancient Greece, the beginning of art history is usually attributed to Giorgio Vasari's "Lives of the Most Excellent Painters, Sculptors, and Architects."[73]. This work, often simply called the "Vitae", marked the first time the lives of painters and their work were described in a systematic fashion, in a work that was entirely dedicated to the history of artists. Since Vasari's seminal work, the approach of analyzing an artist's work along biographical lines has stayed at the heart of art history. As a final goal of this dissertation, we retrace this approach by applying our analysis to different subsets of a bigger dataset, focusing on individual artists or schools of related artists. These analyses will also highlight some interesting strenghts and weaknesses of the representations and methods involved. The shortcomings can mostly be explained by the lack of supervision, pointing to possibilities of improving the quality of stylization results.

1.3 Summary of Contributions

In summary, this dissertation makes two contributions.

The first contribution is the introduction of an *archetypal style representation* based on archetypal analysis. Building on current neural representations for style, we demonstrate that a more interpretable representation can be learned in an unsupervised fashion. This allows the analysis of and reasoning about artistic style by humans on two levels:

First, analyzing the styles in a collection allows for more structured exploration of the collection. Comparing one artwork to others, grouping and decomposing their styles, is an efficient way of building a *mental* model of artistic styles. This can not only be helpful for large collections of artworks; analyzing smaller datasets can help mentally structuring the works of single artists or smaller groups of artists as well. We demonstrate the application of our analysis to several artists and

schools of artists, highlighting the benefits and drawbacks of our method of analysis and the underlying representations.

Second, the archetypal style representation also allows for intuitive control over style manipulation, which can simplify the work with neural style methods. Since the archetypal styles explain a maximum of the styles present in a collection, using these — comparatively few — styles for manipulation can be less overwhelming than choosing from a practically infinite collection of possible (combinations of) style examples. We demonstrate multiple ways in which this manipulation can achieve results unlike other style manipulation methods.

As a second, auxiliary contribution, we modify the stylization procedure of the underlying stylization method "Whitening and Coloring Transform" (WCT) [44] to allow for *small* changes of style, trading off strength of stylization for preservation of detail.

The direct loss of detail in WCT stylization does not always pose a problem in style *transfer* since the goal is often to stylize a natural image in a very notable way. Preserving too *much* detail is in fact a common shortcoming of some style transfer methods. It is thus important to note that the loss of detail exhibited by WCT stylization should not be regarded as a shortcoming per se. To achieve different artistic purposes however, it can be useful to preserve more detail from the content image. This happens to be the case in our specific context, since many of the stylization operations we perform are actually applied to artworks rather than natural images. These images often show much less detail than natural images already; stylizing them to the fullest extent tends to produce less appealing results. That being said, our parameterization offers a more nuanced way to control the stylization process and may be of interest when performing stylization of natural images for artistic purposes.

We have laid out the motivation and challenges of our work. In the following chapter, we will provide further context to for it, discussing the fields and lines of work relevant to ours. Following this, we present our method and results using unsupervised method in chapter Chapter 3. Chapter Chapter 4 offers concluding remarks and outlook.

Chapter 2

Related Work

In this chapter we will discuss the methods that our work relates to. Section 2.1 will lay out the representations used in artistic style manipulation and methods which are typically applied to them. It will become clear that while there have been multiple lines of work on the subject, the most popular one these days is fundamentally based on insights that back well into the last century. It is the combination of these insights with modern computer vision techniques that allowed artistic style manipulation to make the leap from professional movie production settings into modern commodity hard- and software products. Section 2.2 then gives a brief introduction to the learning method we use to navigate these representations.

2.1 Artistic Style Manipulation

Our work concerns the analysis of style in collections of artworks, as well as the manipulation of style using natural images and artworks. It is thus important to understand in detail the problems and solutions previously proposed for both of these tasks. Below, we provide an overview of the tasks and methods discussed in the literature, going into more detail for the works that directly relate to ours.

2.1.1 Texture Generation

The task of artistic style transfer is closely related to the slightly older task of texture synthesis, in which a texture is to be generated that visually resembles an example. Indeed, the tasks of style transfer and texture generation mainly differ

in that a texture does not have a scene composition, but is assumed to depict a material that a human would describe as a single concept. This is not to say that textures do not contain individual objects; as an example, a texture containing many individual pebbles would still be considered homogenous in that it depicts a "pebbled ground". This task of texture synthesis became important when computer graphics became sufficiently advanced that video games and simulators could not reasonably color different objects in flat or shaded colors only. The addition of high quality textures to surfaces dramatically increases the potential for immersion in a video game, and as such has a direct impact on its commercial viability. Creating good textures by hand however, especially for arbitrary geometries, is a difficult and time-consuming task.

Works from the 1990s, such as [25] and [59] treat texture synthesis as a sampling problem: given an example texture, first some image-level statistics are computed. In the case of [25] these statistics are based on wavelet filter responses at different scales. The synthesis then involves sampling from the distribution of images which match the statistics of the example. This approach of modeling textures in terms of image level statistics was based on findings from the 1960s, such as the experiments conducted in [38] that tested which types of texture pairs can and cannot be easily distinguished by humans.

A different way of sampling is chosen by Efros and Leung [14]. They sample the pixels one by one, taking into account the previously sampled pixels. On top of producing convincing results for texture generation, this method lends itself very well to another problem: it can solve the in-painting problem — credibly filling missing pixels in an image — by sampling the missing pixels one by one, conditioned on the pre-existing ones. More importantly for us though, the work has lately been re-visited in its original context of texture synthesis. One of the shortcomings of the original method by Efros and Leung is that it tends to produce repeating patterns that are easily spotted by humans, repetitions that are not in line with the example texture.

In 2015 though, Gatys *et al.* presented their method for texture synthesis [19]. Breaking with the non-parametric approaches, they build an *explicit*, parmetric representation of a texture by exploiting the powerful representations learned by a deep convolutional neural network (CNN). These networks had recently started getting significant attention after the breakthrough results of Krizhevsky *et al.* [41] in the ImageNet Large Scale Visual Recognition Challenge 2012 [61] and had previously been used to classify artworks according to artistic style labels [39]. Gatys *et al.* used these these features as a replacement for the hard coded wavelet features of works as used in [25]. These deep representations, trained to aid other computer vision tasks such as ImageNet classification [11], tend to capture aspects of natural images that correspond to how humans perceive them. Their first layers

learn simple detectors for edges and blobs, often similar to Gabor filters. The higher layers of a CNN also encode more complex semantic features though, features that are no longer directly comparable to gabor filters. For an input image $I \in \mathbb{R}^{H \times W \times 3}$, the (CNN) feature extractor e_l for layer l produces p_l feature maps of size $h_l \times w_l$, with h_l, w_l being the height and width of the resulting feature maps. For neural style applications, it makes sense to represent all the feature maps in a tensor of shape $h_l \times w_l \times p_l$. Collapsing the spatial dimensions, these features can then be interpreted as a matrix $\mathbf{F} \in \mathbb{R}^{p_l \times h_l w_l}$. Gatys *et al.* use the convolutional part of a VGG-19 network [65] to extract features at the chosen layers. Their formulation then takes the shape of a preimage problem, illustrated in Figure 2.1. Starting from an image filled with noise, they match the feature statistics by minimizing the discrepancy with the target statistics using the L-BFGS-B optimizer[4]. At layer l, the style loss to be minimized is

$$\mathcal{L}_{sl} = \frac{1}{4p_l^2 h_l w_l h_{sl} w_{sl}} \left(\mathbf{F}\mathbf{F}^\top - \mathbf{F}^*\mathbf{F}^{*\top}\right)^2 \tag{2.1}$$

where $\mathbf{F}\mathbf{F}^\top$ is equivalent (up to a scalar factor) to the covariance of the average covariance of the iterate texture's feature maps, $\mathbf{F}^*$ is the corresponding feature map from the example texture, and h_{sl}, w_{sl} are the spatial dimensions of the style features at layer l. The normalization by the feature map sizes h_l, w_l, h_{sl}, w_{sl} is crucial, since it accounts for the different feature map sizes. Combining the losses from the chosen output layers of the network, the final optimization objective becomes

$$\mathcal{L}_s = \sum_l w_l \mathcal{L}_{sl}. \tag{2.2}$$

The result of the optimization is an image that has the general appearance of the texture example, while being visually distinct. Since the optimization only considers *statistics* of the feature activation though, it does not impose a specific composition of elements in the result. However, the method turns out to work quite well for textures cropped from artworks, making it an interesting candidate for further develoment by adding a loss component related to image *content*, thus bridging the gap between texture generation and artistic style transfer.

2.1.2 Neural Style changes the name of the game

Artistic style transfer derives from the field of non-photorealistic rendering. While much of this field was focused on interactive drawing techniques, simulating the physics of the different artistic media, works such as [26, 27] treat artistic style as something provided as user input, in the form of examples. In [26] the example is a single image of a painting, with the task being to find a set of parameters that

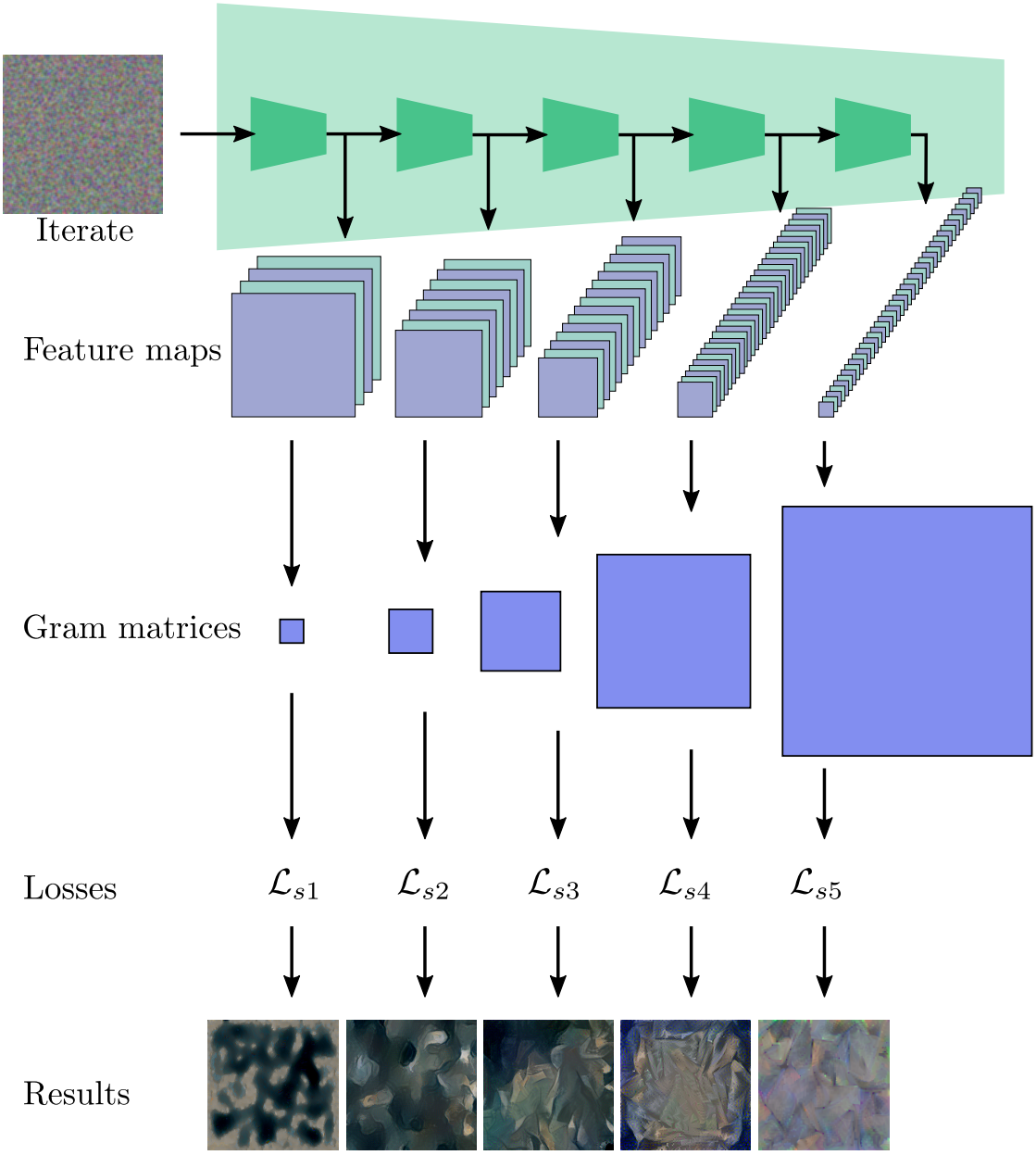


Figure 2.1 – Neural texture synthesis: Features for the iterate (initialized to random noise) are computed at chosen layers of a pretrained neural network. The covariance of the feature maps are averaged across all spatial locations. The loss to minimize for each gram matrix is the mean squared error, relative to the target feature statistics $\mathbf{F}^*$ of the original texture. The images at the bottom show the result of the process using the corresponding layer's features when using Picasso's "Seated Nude" as the style.

allows approximating the style of this painting. In [27] on the other hand the input is a *pair* of images, one natural image, and one artistic image showing the same content. The goal in this setup is to learn the correspondence between the natural image and the artistic rendering which then allows the application of the same style to unseen natural images.

The seminal work of Gatys *et al.* [18] approaches this problem of artistic style transfer by building on their earlier work on texture synthesis. It combines the fact that an artistic style can efficiently be modeled by the feature statistics of a CNN with the ability to approximately reconstruct an image given its CNN feature activations. Since the latter task readily lends itself to a preimage formulation of similar shape as Formula 2.1, this is achieved by simply adding a loss term to the optimization objective in Formula 2.2. Given a feature map $\mathbf{F}_c$ computed from an image as input, the "content loss" is the $l2$ distance:

$$\mathcal{L}_c = ||\mathbf{F} - \mathbf{F}_c||_2^2 \tag{2.3}$$

and the total loss is simply a sum of style and content loss, weighted by the hyperparameter λ:

$$\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_c \tag{2.4}$$

. Figure 2.2 shows this schematically.

Notably, the features extracted from the same VGG-19 network are used for both content and style loss computation. This work marks the beginning of a new wave of rapid development in the artistic style transfer literature. Gatys *et al.* themselves improve on the method in a follow-up work [20] in which they propose several practical additions to the method: by computing several gram matrices over different semantic regions of the image, they can transfer a different style to each region. By performing the style transfer not in RGB, but on the luminance channel only, or by matching the color histograms of the style and content images, they can transfer some aspects of the style, but preserve the color information. Finally, a stylization output at low resolution can be used to initialize a high-resolution stylization, allowing for high resolution images to be stylized without the limited receptive field of the VGG-19 network becoming a problem.

Many works improve upon those by Gatys *et al.* in one aspect or another, and today style transfer is not typically performed using this method. However, it still remains relevant. First, it serves as a useful baseline. For lack of objective ways of measuring the of quality style transfer results, visual inspection is still the predominant way of assessing the quality of style transfer methods. Second, the method is very flexible, compared to others. All the hyperparameters that can influence the quality of results can be modified by the user: the scale of the content image is only bounded by the amount of available GPU memory. More importantly

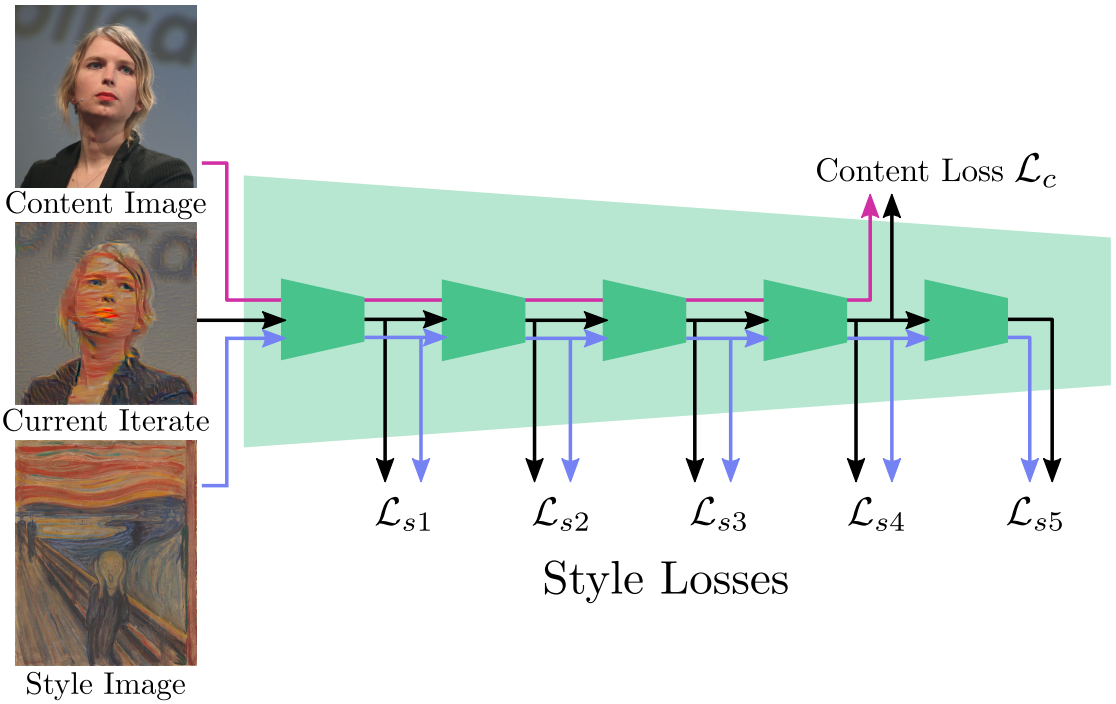


Figure 2.2 – Neural Style: before the optimization, the content and style targets are computed at their respective layers. Every iteration, the content and style features are computed for the current iterate and compared to these targets. This way, the result of the optimization maintains the content features of the content image, while exhibiting feature *statistics* of the style example.

though the scale of the style image, which influences which aspects of the style will be captured by which layers of the netork, can be adjusted within the bounds given by GPU memory and the architecture of the feature extractor. On top of this, all the w_l as well as λ can have a significant impact on the results. The addition of total variation loss can improve results for some use cases. The initialization of the iterate, either as the content image itself, as any sort of random RGB image, or a mixture of the two, allows for different optimization results given the same hyperparameters. The hyperparameters of the L-BFGS-B algorithm itself, mostly the iteration count, may have to be adjusted for some inputs. Of couse, choosing different target layers for either content or style losses will change the results. All these options are left to the user to adjust, which makes the method a powerful tool for enthusiasts and visual artists. They also constitute one of its drawbacks though: for most input pairs, at least *some* of these controls will *have* to be adjusted to achieve a visually pleasing result. This makes the method hard to use for beginners, and makes the production of visual art with it very cumbersome. This is aggravated

by another serious limitation of the method: for each input, a full run of L-BFGS-B has to be performed, which takes on the order of minutes on modern hardware with moderate input sizes. Most of these aspects of the style transfer method have gotten attention in their own right in the meantime. It is worth noting however that most works building on [18] do adopt their *representation* of style, if not directly then often indirectly by using it as a training target.

2.1.3 Addressing the issues with Neural Style one by one

The computational load of the optimization based neural style transfer was among the first issues to be addressed in follow-up work. To overcome the problem of exceedingly slow stlyization, Johnson *et al.* [36] trade off some of the versatility of Gatys' neural style against speed of stylization. They choose a different, though related, network arcitecture, namely the VGG-16 architecture. In principle, they use the same objective as [18] in that they try to produce an image that minimizes the combined content and style loss function. The difference is that while Gatys *et al.* do so by means of optimization, Johnson *et al.* train a feed-forward network to perform the stylization in a single forward pass. Figure 2.3 shows an overview of the method.

For an arbitrary but fixed style S, the stylization network ([36] uses a ResNet) is trained to transform an input image I^c into a stylized image I^s. The transformed image is then fed into a feature extraction network, and style and content loss are computed as in Formula 2.4. Following the image processing literature such as [1] the authors also add a total variation loss term that encourages the network to produce piecewise smooth images. Empirically, the authors choose a set of layers for the losses that produces appealing results: the content loss is applied at layer $ReLU2\_2$, and the style loss is applied at layers $ReLU1\_2, ReLU2\_2, ReLU3\_3, ReLU4\_3$. After the network is trained, it is able to stylize any input image in a single forward pass, allowing real-time style transfer at moderate input sizes on modern hardware. For obvious reasons, this method is known as "fast neural style" as speed at inference time is its main contribution. However, the drawbacks of this method are also clear: apart from the resolution of the content images, all the hyperparameters of [18] have to be chosen at *training* time, leaving the user with a much reduced set of controls. Most importantly of course, the choice of the style input in [36] is also a training time parameter. In this setting, a trained network can only stylize images in one single style. That means that choosing a new style requires re-training an entire network, and that the in-memory representation of a style is as big as all the parameters within the network, which in turn means that storage and transfer of trained styles is becomes a practical problems in domains such as mobile applications. It is worth noting that Ulyanov *et al.* [71] presents a very similar method. There too, the authors train one network per style, to

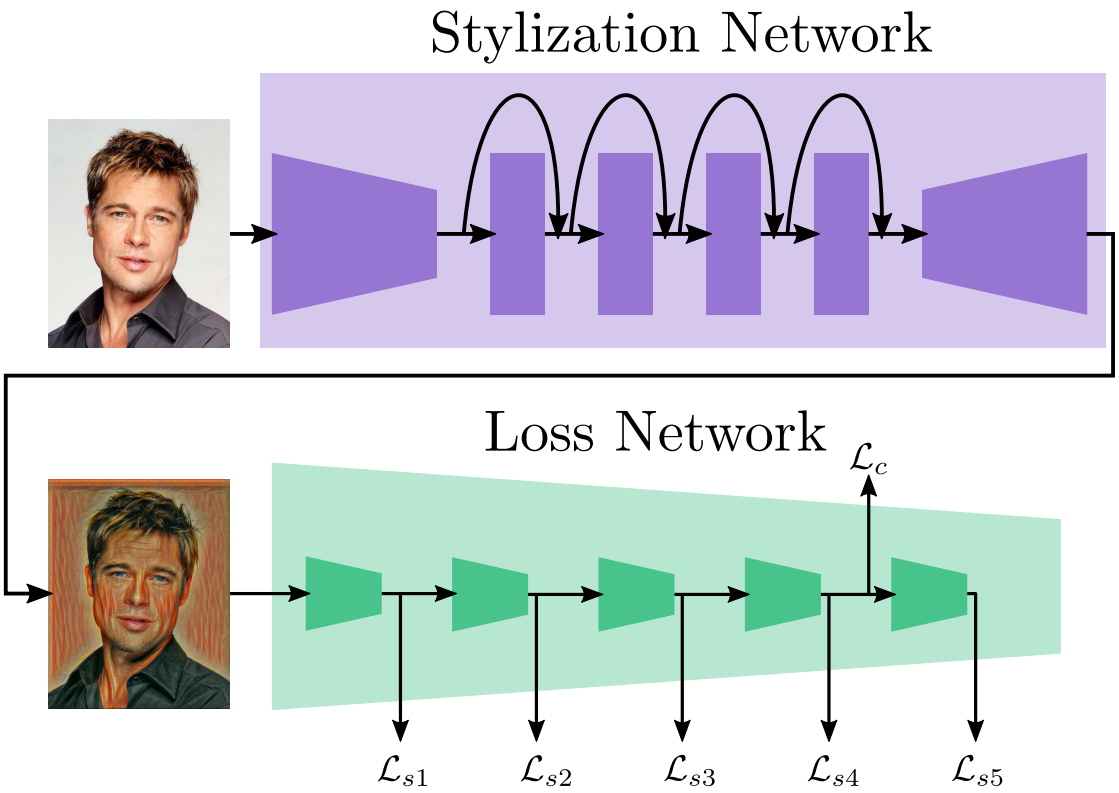


Figure 2.3 – Fast Neural Style: A ResNet is used to transform the input image. It is typically comprised of a convolutional part, multiple residual blocks, and a deconvolutional part. The resulting image is then evaluated according to the style and content loss as defined by Gatys *et al.* Note that the target values for all the loss terms are evaluated ahead of time on a single style example, and so the style image itself is not part of the computation while stylizing an image.

perform stylization in a feed-forward manner. Their primary motivation however is texture generation and their transformer network takes noise tensors as inputs. It transforms the smallest tensor by means of convolutions and nonlinearities, then upsamples and concatenates it with a noise vector at the higher scale. This process is reapeated until the resulting tensor is of the desired size, at which point it is transformed into RGB space, again by a trained convolutional block. From the resulting image a feature extraction network computes the losses in the manner described above. Ulyanov *et al.* extend this method to also perform style transfer. In line with their texture generation procedure, they inject noise vectors into every convolutional block of their generator network. This offers a simple and fast means of resampling different stylization results very quickly. Additionally, by rescaling the noise vectors, the different feature scales can be influenced during stylization.

In the meantime, neural style representations are used for other tasks. For example, [51] classifies artworks according to style or artist using the gram matrix representation from [19]. The authors first extract the features from the same layers as Gatys *et al.*, and compute the feature covariances in the same way. To reduce computational load, they perform a principal component analysis and only use the first up to 4096 dimensions. They then investigate the performance of linear classifiers trained on these representations. This is done separately for all the selected feature layers and their results show that computing the feature covariances of higher layers works better for their task than lower layers.

The preimage approach chosen in Gatys *et al.* proves to be applicable to other image manipulation problems too. In a remarkable demonstration of the power of learned feature spaces, [72] uses the same optimization based preimage formulation to modify the *content* of an image. This is done by computing features of the original image, then displacing those features in a direction related to the desired manipulation, and finally inverting the representation by means of optimization. As a result, semantic modifications like changes of hair color, eyewear, or gender can be achieved.

After presenting their method for feed-forward style transfer [71], Ulyanov *et al.* investigate the influence of normalization layers on the quality of style transfer [69, 70]. Making one simple change to the architecture of their generator network — replacing all batch normalization layers with instance normalization layers — produces much more appealing stylization results. While batch normalization [31] transforms feature maps such that they exhibit zero mean feature activation and unit variance averaged over a *population*, instance normalization transforms each feature map individually, resulting in zero mean and unit variance over the *spatial extent* of the feature map. Given the feature maps $\mathbf{F}_l$ of shape $h_l \times w_l \times p_l$, again interpreted as a matrix of size $p_l \times h_l w_l$, the mean and variance $\mu_l, \sigma_l^2 \in \mathbb{R}^{p_l}$ are computed over the spatial dimensions h_l and w_l. Dropping the index l for legibility, and using element-wise operations only:

$$\boldsymbol{\mu} = \frac{1}{hw} \sum_{j=1}^{hw} \mathbf{F}_{[j]} \tag{2.5}$$

$$\boldsymbol{\sigma}^2 = \frac{1}{hw} \sum_{j=1}^{hw} \left(\mu - \mathbf{F}_{[j]}\right)^2 \tag{2.6}$$

The normalized feature maps are then computed as

$$\widetilde{\mathbf{F}}_{[j]} = \frac{\mathbf{F}_{[j]} - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}} \tag{2.7}$$

where ϵ improves numerical stablility. The authors argue that this more closely matches the way contrast ought to be transferred from the style image to the result.

Additionally, this normalization layer is conceptually and practically simpler than batch normalization since it has no learnable parameters and is applied equally at training and inference time.

Using the instance normalization layer as a basis, Dumoulin *et al.* [12] push feed-forward style transfer further towards the versatility of optimization based stylization. With their method, a network can learn representations for a fixed but arbitrary number of styles. This is achieved by adding another affine transform after each normalization step. Each of these transforms uses learned parameters based on the style to use. For each style s, the authors introduce the parameters $\boldsymbol{\beta}_s$ and $\boldsymbol{\gamma}_s$ that take the inverse roles of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ respectively. The transformed features are then computed as

$$\widetilde{\mathbf{F}}_{[j]} = \boldsymbol{\gamma}_s \left(\frac{\mathbf{F}_{[j]} - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}} \right) + \boldsymbol{\beta}_s. \tag{2.8}$$

Dumoulin *et al.* name this transform *conditional instance normalization*. This encapsulates the effective representation of each style into fewer parameters than before. For L layers with p_l feature maps each, the total representation specific to a single style is only of size $\sum_{l=1}^{L} 2p_l$. The rest of the parameters in the stylization network are shared between styles. This is in contrast to [36] where each style requires storage of a full set of network weights. It is at this point that storing and distributing multiple styles on mobile devices really becomes feasible; consumers can choose from a fixed set of styles, much like the predefined image filters that modern photo sharing apps provide.

In a similar vein, Chen *et al.* [5] train an encoder/decoder architecture with conditional operations per style. Like [12], their method, called StyleBank, is also able to stylize images with a fixed but arbitrary number of styles. While conditional instance normalization uses an affine tranform for each style, StyleBank uses a more powerful convolution operation, but only applies it at the bottleneck layer of the encoder/decoder transformer network. On top of the change in feature transform and architecture, the authors also add an autoencoding loss term to the losses used in [36]. Encoder, decoder, and the StyleBank convolutions are all jointly trained. However, the method allows to train additional styles later while keeping encoder and decoder fixed.

A different way of aligning the features of a network's hidden layers is proposed in [6] [1]. As is the case for StyleBank, Chen and Schmidt train an autoencoding network and perform an alignment operation in the bottleneck feature space. Instead of transforming the content feature maps given statistics of the style image however, Chen and Schmidt treat the alignment step as a sampling problem. They sample overlapping patches of both content and style feature maps, and then

1. Note that [5] and [6] are authored by *Dongdong Chen* and *Yuansi Chen* respectively.

produce a new feature map by replacing each content feature patch by its closest match from the style feature patches. Recombinging these overlapping patches by averaging the overlapping sections yields feature maps that share characteristics of both the content and style image. This "style swap" operation, as the authors call it, is amenable to an efficient implementation with standard CNN operations. However, it does result in a slight domain shift between unmodified style feature maps and locally averaged style feature patches. This is addressed by including this domain shifted data in the training data for the transformer network. While the encoder part of the network is chosen to be a VGG-19 network up to and including layer ReLU3_1 and is kept fixed during training, the decoder is built as a mirror image architecture and trained to invert the encoder. The decoder is trained with a re-encoding loss, and a total variation prior in the RGB space. That is, for the (fixed) encoder e, the decoder d_θ with parameters θ, and the style-swapped features $\mathbf{F}$, the loss is

$$\mathcal{L}(\theta) = ||\mathbf{F} - e\left(d_\theta\left(\mathbf{F}\right)\right)||_F^2 + \lambda \mathcal{L}_{TV}\left(d_\theta(\mathbf{F})\right) \tag{2.9}$$

where $||\cdot||_F$ is the Frobenius norm.

These works — StyleBank, conditional instance normalization, and style swap — share the idea of replacing the style loss term of Gatys *et al.* by a modification in feature space. But while StyleBank and conditional instance normalization already present significant improvements over training one network per style, they lack the versatility of style swap since they require training for each style to be produced. Style swap allows for what is commonly called *arbitrary* style transfer. Also called "zero-shot style transfer", this setting brings back the flexibility from [18] that was traded off for speed with fast neural style. To "solve" arbitrary style transfer, a method should be able to credibly transfer any style to any content. This is a high bar to set, and for many style/content pairs, it is not even clear to humans what a credible results should look like. Naturally, there are multiple lines of work approaching the problem from different angles. Most of these works follow a scheme involving three processing steps that are also readily recognized in the three aforementioned methods:

1. The content image is encoded into an intermediate CNN feature space.
2. A new set of feature maps is created as a function of the computed content features, as well as some representation of the style image. This often (but not necessarily) includes computing intermediate features for the style image as well.
3. The resulting features are decoded to an image that is the result of the process.

The type of feature modification, the nature of the encoding and decoding networks, as well as the way these networks are (or are not) trained, are thus the points by which the different methods are distinguished.

One well known method falling into this category is *Adaptive Instance Normalization* (AdaIN) [30]. Building on conditional instance normalization, this work proposes to simply apply affine transforms to the content features so that after transformation they have the mean and variance of the style features. This is done by training the network with random pairings of content and style images. For any layer at which AdaIN is to be applied, this means first applying regular instance normalization to the content features obtained at that layer. To these normalized features, the conditional instance normalization operation as seen in Formula 2.8 is then applied, with $\boldsymbol{\beta}_s$ and $\boldsymbol{\gamma}_s$ computed from the style features at the same layer.

It is worth noting that the architecture introduced in [30], namely the use of affine transformations in many successive steps, has successfully been adapted as "StyleGAN" for natural images in [40]. This modification of previously existing GAN generator architecures marked a significant step up in the quality of results.

Another direct extension of [12] to allow for arbitrary style transfer is [21]. In contrast to [30], here the parameters $\gamma_{\mathbf{s}}$ and $\beta_{\mathbf{s}}$ are not learned per style and fixed after training. Instead, a second branch of the network is trained to *predict* suitable $\gamma_{\mathbf{s}}$ and $\beta_{\mathbf{s}}$. The two networks — style parameter prediction and transformer network — are jointly trained to minimize the classic combined style and content loss based on a VGG network, as is the case for most feed-forward neural style methods. The network predicting the normalization parameters uses an Inception-v3 [67] architecture, pretrained for image classification, on top of which two fully connected layers are trained. The second of these is the output layer computing the normalization parameters. The first of these top layers is deliberately chosen to have only 100 hidden units, wich allows an analysis of style representations in $\mathbb{R}^{100}$.

The Whitening and Coloring Transform (WCT)

Another important member of the feature modification family of methods is [44] titled *Universal Style Transfer via Feature Transforms*. This method is most often referred to as WCT, with its namesake, the *Whitening and Coloring Transform*, being at the heart of the method. The method is remarkably simple, especially the encoding and decoding parts: The encoder is fixed to be the first few convolutional layers of a normalized VGG-19 network pretrained for image classification, one of the most common choices for neural style manipulation. Following a good number of works in the field, the authors choose the convolutional layers $ReLUX\_1$, $X = 1, \ldots, 5$ as target layers for transfer. Given an encoder, the corresponding decoder is trained as a simple feature inverter. This is done using a reconstruction loss and a re-encoding loss. That is, given a batch of image *patches* $\mathbf{X}$ and a feature extractor e_l that extracts features at layer l, the decoder d_l is trained to reconstruct the image patches as $d_l(e_l(\mathbf{X}))$ by optimizing

$$\min_{\theta_{d_l}} ||\mathbf{X} - d_l\left(e_l\left(\mathbf{X}\right)\right)||_2^2 + \lambda||e_l\left(\mathbf{X}\right) - e_l\left(d_l\left(e_l\left(\mathbf{X}\right)\right)\right)||_2^2. \tag{2.10}$$

Even the training of these feature inverters is done on the Coco dataset [45], which means that both feature extractors (VGG-19, trained for ImageNet classification) and feature inverters (trained on the Coco dataset) are trained on natural images exclusively; they hold no information about artistic images or their style. Furthermore, since their training does not depend on style examples, they are trained *once* and are kept fixed afterwards.

The feature *manipulation* stage then involves no learning at all, and gives the method its colloquial name. The *Whitening and Coloring Transform* (WCT) changes the features extracted from the content image to have the same mean feature activations and covariances across spatial dimensions as those extracted from the style image.

To perform stylization at layer l given an encoder e_l and a decoder d_l trained as described above, first the content features $\mathbf{F}_l^c = e_l(I^c)$ and $\mathbf{F}_l^s = e_l(I^s)$ are computed. These will be tensors in $\mathbb{R}^{h_l \times w_l \times p_l}$ and can be considered as matrices in $\mathbb{R}^{p_l \times h_l w_l}$ as above. We will drop the index l again, to preserve legibility. The whitening operation is then defined as

$$\widetilde{\mathbf{F}^c} := \mathbf{W}^c(\mathbf{F}^c - \boldsymbol{\mu}^c) \tag{2.11}$$

where $\boldsymbol{\mu}^c$ is the mean feature activation of the content feature maps, i.e. a vector in $\mathbb{R}^{p_l}$, and $\mathbf{W}^c$ is a whitening matrix that decorrelates the features. It is computed from the eigendecomposition of the content features' covariance matrix. If

$$\mathbf{F}^c\mathbf{F}^{c\top} = \mathbf{E}^c\mathbf{D}^c\mathbf{E}^{c\top} \tag{2.12}$$

where $\mathbf{E}^c$ are the eigenvectors of $\mathbf{F}^c\mathbf{F}^c$ and $\mathbf{D}^c$ is a diagonal matrix holding the corresponding eigenvalues, then

$$\mathbf{W}^c = \mathbf{E}^c\mathbf{D}^{c-\frac{1}{2}}\mathbf{E}^{c\top}. \tag{2.13}$$

This operation effectively removes the style information as understood by Gatys *et al.* from the content features.

The coloring operation is the conceptual opposite of the whitening operation. It uses the mean feature activation of the style feature maps $\boldsymbol{\mu}^s$ and the coloring matrix

$$\mathbf{C}^s = \mathbf{E}^s\mathbf{D}^{s\frac{1}{2}}\mathbf{E}^{s\top} \tag{2.14}$$

where $\mathbf{E}^s$ and $\mathbf{D}^s$ are the eigenvectors and eigenvalues of the style features' covariance as described above for $\mathbf{E}^c$ and $\mathbf{D}^c$.

In the following, we simply summarize the entire WCT operation as a single function $C^s : \mathbb{R}^{p \times m} \to \mathbb{R}^{p \times m}$ which can be written as

$$C^s(\mathbf{F}_c) := \mathbf{C}^s \mathbf{W}^c (\mathbf{F}^c - \boldsymbol{\mu}^c) + \boldsymbol{\mu}^s \tag{2.15}$$

The resulting feature maps maintain most of the spatial structure of the content features, but exhibit local characteristics of the style features in the form of their feature covariances. To control the strength of the stylization, the authors propose to interpolate along the line from the original content features and the transformed features using the hyperparameter γ. Inverting the resulting features with the corresponding decoder d_l yields the stylization result for layer l:

$$\hat{I}_l = d_l \left(\gamma C_l^s (e_l(I^c) + (1 - \gamma) e_l(I^c)) \right) \tag{2.16}$$

where $\gamma \in [0, 1]$.

Assuming that the feature transform does not place the modified features too far outside the support of the decoder, this method produces an image that preserves the content of I^c, while producing feature activations with covariance similar to $\mathbf{F}_s$ upon encoding with e_l. This image is then used as the input for the next stylization step with encoder e_{l+1} and decoder d_{l+1}. The sequence of layers at which to transfer is chosen on the assumption that encoders with more layers will introduce more abstract style elements into the image, while encoders with fewer layers modify predominantly colour and texture of the image. This assumption, based on previous analysis of the features learned by CNNs, as well as the size of their receptive fields, is experimentally validated in [44]. The authors thus decide on a stylization procedure starting at layer $ReLU5\_1$, and finishing with layer $ReLU1\_1$, as shown in Figure 2.4.

Naturally, follow-up works have tried to improve on WCT in various ways. Among these, [77] is notable for considering not only the covariance of features of a given layer, but also their covariance with features from other layers. Scaling the later feature maps up to the size of preceding ones allows the computation of these *inter-layer* feature covariances.

A method incidentally related to [44], but independently developed as an extension of [30], is [79]. Instead of computing the target features by means of the (computationally expensive) matrix square root and inverse square root operations, the authors suggest to treat the entire transform as part of the learning problem. The feature modification then becomes

$$\mathbf{F}^{cs} = \mathbf{W} \mathbf{F}^s \mathbf{F}^{s\top} \tag{2.17}$$

where $\mathbf{W}$ is a trainable matrix in $\mathbb{R}^{p_l \times p_l}$. While this method is not able to learn the full WCT transform, it can produce acceptable approximations, or at least

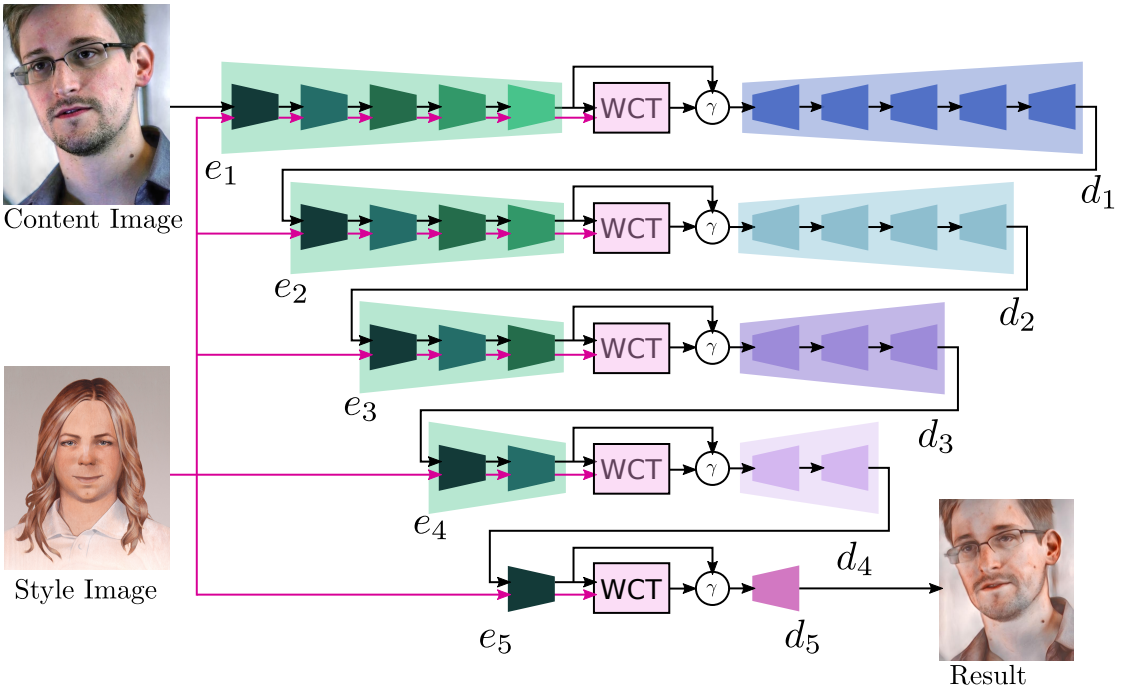


Figure 2.4 – WCT style transfer as performed by Li et al. For each encoder / decoder pair, features are computed both for the current content image and the style image. The content features are then transformed by whitening and coloring to have the covariance of the style features. The hyperparameter γ controls the strength of the stylization. Finally the modified content features are decoded back into RGB space, and the process is repeated with the newly-styled image as the new content input.

useful results. By sacrificing this exactness of WCT, the transfer becomes much less computationally intensive, at the cost of more training load.

Another interesting application of the WCT is presented in [64]. "Avatar-Net" uses the same whitening as [44] to whiten patches of content and style feature maps before matching them in the fashion of StyleBank [5]. The coloring transform is then used to produce the final modified features to be decoded. The authors argue that this allows this feature modification to better match the distribution of the style features, resulting in better transfer results.

One problem that WCT transfer exhibits is that it often stylizes one semantic area with a mixture of style elements present in the style example, when using a single texture for the whole area would have been more appropriate. Figure 2.5 shows an example of this behaviour: even though both the content and style image can roughly be divided into sky, fields, and road, with a smattering of trees and bushes, the algorithm makes no note of these semantic regions and transfers

Figure 2.5 – A failure case of WCT. Even though content and style image depict very similar scenes, the method applies styles in the wrong semantic regions.

patches of field texture into the sky and pieces of sky onto the road. A follow-up to WCT, [57], addresses this problem by identifying different *substyles* in the style image, and automatically matches them to semantic regions in the content image. This allows for much more believable transfer results.

2.1.4 Parallel and intersecting lines of research

There is another line of research that has been developed in parallel to the neural style methods. These works build directly on the sampling based approach in [14] and [15], but improve on them to address their shortcomings. The more recent works in this line, such as [17] and [66], add guidance information to the stylization process. This allows to distinguish between regions with similar appearance, but different semantics and thus texture. For [17] and [66], this guidance information is mostly obtained from a known scene geometry, but later works allow for slightly less involved guidance information. In [33], a video is styled by manually stylizing one or more key frames only.

The quality of their results is remarkable, and it is easy to see that this mode of operation fits very well with professional media production workflows. This is a setting in which huge numbers of frames need to be stylized, and providing standardized style templates (simple scenes with a known geometry, painted in the desired style) is a reasonable requirement to integrate the technique into a production workflow. It is however a very different level of involvement from most neural style methods, where the user is expected (and expects) to simply provide a style example that is not necessarily semantically related to the content image, or even just to choose from a few pre-selected styles presented by an application. As such, the work on methods involving guidance information are not directly compatible to most neural style methods. However, recent work manages to join the two lines of research: [68] uses a neural style transfer method to generate a low-resolution stylization, then improves upon it by means of patch based transfer.

Another line of newer research on style transfer, which is a bit more closely

related to neural style, is based on Generative Adversarial Networks (GANs). Introduced in 2014 [23], adversarial learning has been quickly adopted to address different tasks. The idea of the method is to train one network (the *discriminator*) to compute a useful training signal to another network (the *Generator*). In the most general terms, the task of the generator is to provide data; that is, it has to transform a known distribution — and one that is easy to sample from — into the distribution of the training data. The task of the discriminator is to distinguish between generated and real data.

The overall learning objective is then

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})} [logD(x)] + \mathbb{E}_{z\sim p_{\mathbf{z}}(\mathbf{z})} [1 - log(D(G(\mathbf{z})))] \tag{2.18}$$

where p_{data} is the (unknown) distribution that is assumed to be generating the real data, and $p_{\mathbf{z}}$ is a known distribution, usually a multivariate normal distribution. GANs — and especially their later variants such as Wasserstein GANs [2, 24], or networks using spectral normalization [52] — have been applied to datasets of artworks with varying degrees of success. In fact, one of the most well-known — and controversial — applications of a GAN to date was presented in 2018 by the art collective *Obvious*. A canvas print of a portrait of "Edmond de Belamy"[13] was created by training a GAN on a collection of 15,000 portraits, and was sold at the auction house Christie's for a record breaking sum. Training the model simply on unlabeled images puts this work into the same setting as academic works such as DCGAN [60] and works building on it like [37, 16], which was part of the controversy around the auction, and also puts the work firmly outside the style manipulation setting, since it is not possible to control the content of the generated portraits by providing examples at inference time.

More relevant for style manipulation are works that tackle image to image translation tasks. First, pix2pix [32] showed that conditional GANs can be used to solve paired translation tasks, such as aerial imagery and maps. Conditional GANs, as their name implies, use distributions that are conditional on some known data. In the case of [32] these known data are the images in the source domain. This setting however is not applicable to style transfer since the style images do not depict scenes that are also available as natural images. However, CycleGAN [81] removes the requirement for paired image pairs, by adding a cycle consistency loss. By requiring that the translation result can be translated back to the source domain all the while remaining indistinguishable from a real data point, a pair of translation networks can be trained to perform translation between two domains without requiring any paired data. As [8] shows, much of this translation process relies on the networks' adding of perturbations imperceptible to humans to the resulting images, perturbations that encode the necessary information for the translation back to the soure domain. Nevertheless, this method is of great interest

for the style transfer task, since it maps very directly to the reasonable setting of training a model to emulate the styles found in a collection of artworks. It does not, however, fit the setting of providing one style input and one content input. Instead, it can only translate from a source domain (that can be the domain of natural images) into the target domain, for which there needs to be sufficient training data. Also, CycleGAN does not work well when the translation is not well defined in at least one direction, since in that case the cycle consistency loss becomes (near-) meaningless. Sadly, this is the case for many more modern artworks which depict no scenes whatsoever and thus have no correspondence in natural images.

There are thus some works that try to accommodate for the specifics of artistic style in one way or another. Addressing the problem of image in-painting, [22] learns a representation of style on the *BAM* dataset [74] (which uses very coarse style labels) by constructing a triplet loss [28] using images with differing content but similar style, and different style, but similar content. These representations are then used for patch *retvrieval*, to aid the ultimate goal of in-painting.

GANs have also been used to an end similar to the patch based transfers of [14, 15, 17, 66, 33, 68]. While these methods sample directly from the style example image, [34] samples large patches from it before performing the actual transfer. The patches are cut and tiled to match the size of the content image, and provided to a transfer network as additioal input channels. The transfer network — trained with an adversarial loss — then computes, for each pixel coordinate, a mixing between the content image at the given location and any number of the sampled patches, also at the location in question. As such, the mixing matrix chosen by the network resembles a sampling process of image patches from the content image and style image alike.

Another GAN based work that tackles the specificities of artistic style is [80], in which the task is a form of super-resolution for textures. This task differs slightly from super-resolution for natural images in that it becomes necessary to faithfully reproduce texture patterns locally, while varying them globally in a manner consistent with the original image.

Most relevant to us in terms of incorporating domain knowledge into the model is the work of Sanakoyeu *et al.* [62]. Their transformer network has a classic encoder/decoder structure with a bottleneck layer. To improve the quality of feed-forward style transfer, they add a loss term that compares the latent codes of the transfer output and a re-encoding of the same output through the encoder. The network is trained with this "style aware content loss" as well as an adversarial loss and an autoencoder loss. The latter is modified insofar that both input and output of the transformer network are first transformed by a common (simple) network to avoid a direct comparison in pixel space.

As has surely become obvious from the above, the field of artistic style transfer

in general, but also that of neural style more particularly, has been quite active in recent years. Many concurrent developments by different actors have quickly moved neural style from the manual, slow, trial and error process of Gatys *et al.* to the fully-automatic style transfer based on [21] which was presented by the team building Google's game streaming platform Stadia [58] and which allows stylization of full screen game content in real-time.

It is worth reiterating however, that nearly all of the methods described above rely, directly or indirectly, on the style representation chosen by Gatys *et al.* as statistics of feature activations across spatial locations. The reasons for this are mostly pragmatic — the representation allows practical applications with results of good visual quality — but have their justification in the early work of Julesz and others.

In our work, we too heavily rely on this representation. But we are less interested in improving the quality of transfer, than in using the representation to navigate a *collection* of artworks. For this, we learn a representation *on top of* the representation of Gatys *et al.* The next section describes tools we use to learn this representation

2.2 Dictionary Learning

To be suitable for exploring and using collections of artworks, a method must first and foremost deliver *useful* results. For our intents, to be useful, results must be interpretable, since the purpose is for the user of the method to gain an understanding of the styles of individual artworks as well as collections and how they relate to one another. We thus aim to find a representation of style that has *meaningful* dimensions and only presents few concepts at a time to a user, since humans can only keep track of few concepts at a time.

It is clear that the representations used to perform style *manipulation* do not meet these requirements in the least. The full gram matrices computed by [19, 18] have hundreds of thousands of dimensions. Even the representation of [21] with it's 100 dimensions is incomprehensible to a human. However, the latter reperesentation is smooth enough to allow for simple visualizations of directions in the 100-dimensional space. That hints at the possibility of learning meaningful directions in the space of style representations, or at least locally meaningful ones.

That leaves us with the question which method to use to best be able to interpret the results. These considerations lead us directly to the concept of parsimony in machine learning models. Parsimonious methods try to explain a maximum of the available data with a minimum of modeling. A well established method parsimonious modeling is dictionary learning. In dictionary learning, a set of training samples is explained by reconstructing each sample as a linear combination

of *atoms*. For data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, a dictionary $\mathbf{D} \in \mathbb{R}^{p \times k}$ with k entries is learned by solving

$$\min_{\mathbf{D} \in \mathcal{C}, \mathbf{A} \in \mathbb{R}^{k \times n}} \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda \boldsymbol{\Psi}(\boldsymbol{\alpha}_i) \tag{2.19}$$

where $\mathbf{A}$ holds the $\boldsymbol{\alpha}_i$ as its columns. That is, the columns $d_j \in \mathbb{R}^p$ of dictionary $\mathbf{D}$ occupy the same space as the data points $\mathbf{x}_i$. They are typically constrained to the unit sphere:

$$\mathcal{C} = \{\mathbf{D} \in \mathbb{R}^{p \times k} : \forall j ||\mathbf{d}_j||_2 \leq 1\}. \tag{2.20}$$

The function $\boldsymbol{\Psi}$ in Formula 2.19 is used to induce sparsity in $\mathbf{D}$, that is to encourage the opimization procedure to set as many values of $\mathbf{D}$ as possible to zero. The scalar λ then takes the role of weighing the two objectives against one another. On the one hand, faithfully reconstructing the data is necessary to make the model useful; an inaccurate model of the data will not allow for new insight to be gained. On the other hand however, parsimony is in itself a useful property of the model. This is true for two main reasons: sparsity can often ease the computational burden of a model. In our case however, the fact that sparse models tend to be more interpretable is of the most interest. For a more detailed discussion of sparse modeling in the context of computer vision, see Mairal, Bach, and Ponce [49]. Here we give a brief overview of the method we rely on for our work.

As we have laid out above, interpretability of results is one of our main concerns in this work, and dictionary learning clearly shows some merit in this regard. However, its interpretability has its limits. The entries of the dictionary, though they may allow for accurate reconstruction of and technically reside in the same space as the data, are not necessarily meaningful in themselves. For some types of data and some problems, they may still be readily interpretable by a human, but this is not always the case.

To provide this two-way relationship between data and dictionary we make use of a technique called "archetypal analysis".

2.2.1 Archetypal Analysis

Given a set of vectors $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ in $\mathbb{R}^{p \times n}$, archetypal analysis [10] learns a dictionary $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_k]$ in $\mathbb{R}^{p \times k}$ with added constraints: on the side of the data, each sample $\mathbf{x}_i$ is approximated by a *convex* combination of archetypes — that is, there exists a code $\boldsymbol{\alpha}_i$ in $\mathbb{R}^k$ such that $\mathbf{x}_i \approx \mathbf{Z}\boldsymbol{\alpha}_i$, where $\boldsymbol{\alpha}_i$ lies in the simplex

$$\Delta_k = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^k \text{ s.t. } \boldsymbol{\alpha} \geq 0 \text{ and } \sum_{j=1}^{k} \boldsymbol{\alpha}[j] = 1 \right\}.$$

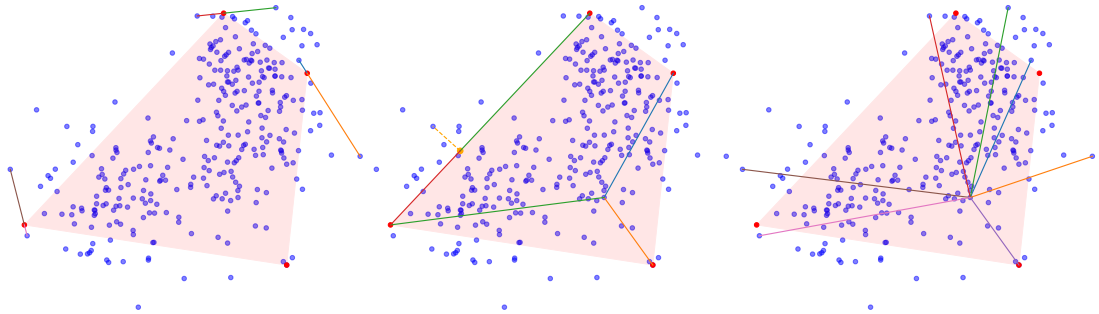


Figure 2.6 – Left: The archetypes (red) are chosen to approximately span the convex hull of the data (blue). They are sparse convex combinations of only few data points. Note that the bottom right archetype is degenerate: it is the trivial convex combination of only one data point. Middle: While data inside the convex hull of the archetypes can be accurately recovered, data outside of it is projected onto the closest point of the hull. Right: Since archetypes are convex combinations of data points, and the data are reconstructed from the archetypes, the reconstruction can be thought of as a convex combination of data points.

Conversely, each archetype $\mathbf{z}_j$ is also constrained to be in the convex hull of the data and there exists a code $\boldsymbol{\beta}_j$ in Δ_n such that $\mathbf{z}_j = \mathbf{X}\boldsymbol{\beta}_j$. The natural formulation resulting from these geometric constraints is then the following optimization problem

$$\min_{\substack{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_n \in \Delta_k \\ \boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_k \in \Delta_n}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}_i\|^2 \quad \text{s.t.} \quad \mathbf{z}_j = \mathbf{X}\boldsymbol{\beta}_j \text{ for all } j = 1, \dots, k, \tag{2.21}$$

which is simply a dictionary learning problem as seen in Formula 2.19 with the extra constraints added. Figure 2.6 gives an intuition for a two-dimensional case.

This optimization problem can be addressed efficiently with dedicated solvers [7]. Note that the simplex constraints lead to non-negative sparse codes $\boldsymbol{\alpha}_i$ for every sample $\mathbf{x}_i$ since the simplex constraint enforces the vector $\boldsymbol{\alpha}_i$ to have unit ℓ_1-norm, which has a sparsity-inducing effect [49]. As a result, a sample $\mathbf{x}_i$ will in practice be associated to a few archetypes, thus allowing for easy interpretation of the results. Conversely, an archetype $\mathbf{z}_j = \mathbf{X}\boldsymbol{\beta}_j$ can be represented by a non-negative sparse code $\boldsymbol{\beta}_j$ and thus be associated to a few samples corresponding to non-zero entries in $\boldsymbol{\beta}_j$. This too increases the interpretability of results.

Despite first being presented over two decades ago, archetypal analysis has remained relatively obscure. Over the years, there have still been some works working on and with archetypal analysis. For example, [3] applies a kernelized version to time series of web search data. In [7], archetypal analysis is applied in the visual domain, specifically on photos of famous tourist destinations. By replacing the dictionary learning step in a traditional bag of visual words pipeline [76] based

on SIFT [47] features, the authors obtain a set of mostly interpretable archetypes summarizing the dataset.

There are a few notable failure cases of archetypal analysis. The first one is that while the sparsity of the solutions *encourages* interpretable archetypes, it by no means guarantees them. Not every archetype will represent one clearly defined concept or aspect of the data. This is especially true if the number of archetypes k is not chosen appropriately. Since with enough archetypes the data is sufficiently well approximated, the reconstruction loss term in Formula 2.21 becomes very small and the sparsity-inducing regularizer of the objective function gains importance for the optimization. Choosing a high number of archetypes will thus result in very sparse solutions, and will in fact regularly produce β_j that only have one non-zero component. But choosing k too small yields bad results too. In that case the algorithm ends up trying to cover more than one extreme point of the convex hull of the data with a single archetype. This makes the archetypes less interpretable since they will be comprised of more than one concept. They also become much less sparse then, since the reconstruction loss term dominates the total objective function. This problem of choosing the right number of archetypes is very similar to the the corresponding problem in clustering algorithms such as k-means. The practical approach to it is usually to manually choose a parameter based on inspection of the results. Typically, there exists a range of values for which the quality of results remains quite stable. In the experimental section of Chapter 3 we discuss all of these problems in the context of our work.

We have now laid out the context of our work and how it relates to prior works, as well as parallel and otherwise related lines of work. Building on this, the next chapter will present the contributions we make.

Chapter 3

Unsupervised Learning for Style Analysis and Manipulation

Contents

The preceding chapters have laid out the motivation for our work and have given context to the tasks at hand. After a survey of the works relating to ours, we discussed the WCT style transfer method [44] which will form the base for most of our work, as well as archetypal analysis [10] which will be our method of choice for the analysis of artworks. As is apparent from the survey of neural style works, the

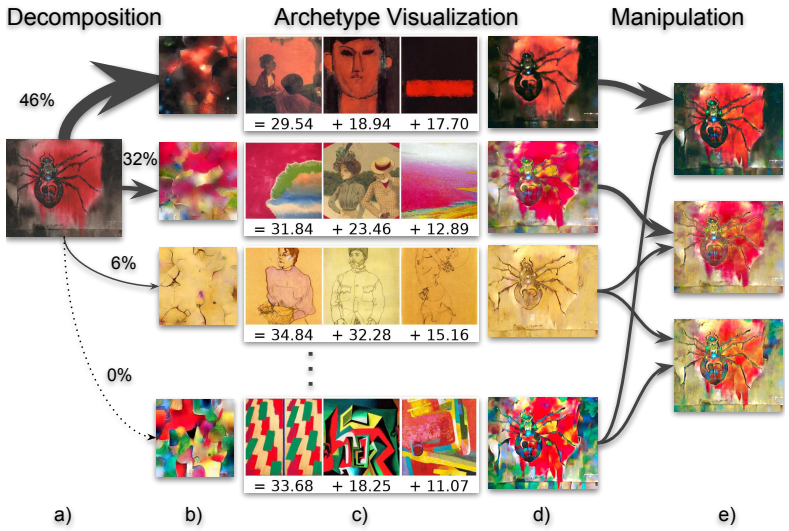


Figure 3.1 – Overview of our method: using deep archetypal style analysis, we can represent an artistic image (a) as a convex combination of archetypes. The archetypes can be visualized as synthesized textures (b), as a convex combination of artworks (c) or, when analyzing a specific image, as stylized versions of that image itself (d). Free recombination of the archetypal styles then allows for novel stylizations of the input (e).

goal of *analyzing* artworks, rather than just manipulating photos with their help, sets us apart from most of the literature, where the goal is to improve the quality or runtime properties of artistic style transfer. Instead, we have the objective to automatically discover and summarize artistic styles present in a collection of artworks. We will now present this method which combines WCT and archetypal analysis, resulting in a new, derived representation of style which crucially lends itself to human interpretation. This makes this archetypal representation of style more suitable to our task of collection and artwork analysis and summary than the representations of style typically used for style *transfer* in the literature. The results of the analysis can then be interpreted either on a collection level, to identify common or uncommon styles in the collection, or on the level of a single artwork, to gain insight into its particular style. Furthermore, we can use this representation as a parameterization for style *manipulation*, given that the underlying style transfer method allows for the interpolation between styles.

Archetypes are simple to interpret since they are related to convex combinations of a few image style representations from the original dataset, which can thus easily be visualized. When applied to painter-specific datasets, they may for instance capture the variety and evolution of styles adopted by a painter during his career, if it is diverse enough.

Moreover, archetypal analysis with its mutual decomposition model naturally allows for a dual interpretation view: On the one hand, archetypes can be seen as convex combinations of image style representations from the dataset. This allows a view at the collection level. On the other hand, each image's style can also be decomposed into a convex combination of archetypes, allowing us to put one artwork's style into relation to others found in the same collection. Then, given an image, we may for example automatically interpret which archetypal style is present in the image and in which proportion, which is a much richer information than what a simple clustering approach would produce. When applied to rich data collections, we sometimes observe trivial associations (e.g., the image's style is very close to one archetype), which is to be expected, since many aspects of artistic style are trivial too, and thus often go without saying. But we also discover meaningful interesting relationships, where an image's style may be interpreted as an interpolation between several archetypes.

We will first discuss archetypal analysis as a natural tool for unsupervised learning of artistic style, and we also show that it provides a latent parametrization allowing to manipulate style by extending the universal style transfer technique of [44]. By changing the coefficients of the archetypal decomposition (typically of small dimension, such as 256) and applying stylization, various effects on the input image may be obtained in a flexible manner. Secondly, transfer to an archetypal style is achieved by selecting a single archetype in the decomposition; style enhancement consist of increasing the contribution of an existing archetype, making the input image more "archetypal". More generally, exploring the latent space allows to create and use styles that were not necessarily seen in the dataset. Figure 3.1 shows as schematic overview of the method.

To the best of our knowledge, [21] is the closest work to ours in terms of latent space description of style; our approach is however based on significantly different tools and our objective is different. In [21] a latent space is learned for style description in order to improve the generalization of a style transfer network to new unseen paintings. In contrast, our goal is to build a latent space that is directly interpretable, with one dimension associated to one archetypal style, and paintings' styles being described by a convex combination of archetypes. We do however use their concise style representation for our experiments, since they have already been demonstrated to be smooth enough to find meaningful directions in their spaces.

The rest of this chapter is organized out as follows: Section 3.1 presents the style representations we use, the archetypal style analysis model and its application to a large collection of paintings. Section 3.2 shows the use of archetypal styles for various style manipulations. We present implementation details and results of both analysis and manipulation in Section 3.3. This includes an investigation into the relationship of metadata *not* used during training and the information that the

archetypal style representation manages to capture despite this.

Note that a shorter version of this chapter was presented before as a conference paper:

Daan Wynen, Cordelia Schmid, and Julien Mairal. *Unsupervised Learning of Artistic Styles with Archetypal Style Analysis*. May 28, 2018. URL: http://arxiv.org/abs/1805.11155 (visited on 05/30/2018)

We extend on this work by introducing a new large-scale dataset based on the WikiArt website containing about 120 000 paintings with annotations, studying additional style representations not considered in the conference submission, and perfoming an analysis of information captured by archetypal styles using the metadata from the WikiArt dataset.

3.1 Archetypal Style Analysis

In this section, we introduce the representations of artistic style we will be using throughout this work. We show how archetypal analysis applied to two different style descriptors leads to a low-dimensional, sparse, and thus often interpretable, encoding of style.

3.1.1 Feature covariances as a descriptor of style

As described in Chapter 2, artistic style is often described as a collection of local statistics of features maps produced by a convolutional neural network [43]. The most classical representation consists in computing feature covariances and this choice is still at the heart of many state of the art style transfer methods. For instance, this is the style description used by the universal style transfer method of [44].

More precisely, given an input image denoted by I, we consider a set of feature maps $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_L$ produced by a deep network. Following [44], we consider the ReLU_X_1 layers of the VGG-19 network [65] which has been pre-trained for classification. However, we do not include the ReLU_1_1 layer into the analysis, since it is very close to the original RGB space and its inclusion usually introduces a strong focus on color during analysis. Each feature map $\mathbf{F}_l$ may be seen as a matrix in $\mathbb{R}^{p_l \times m_l}$ where p_l is the number of channels and m_l is the number of pixel positions in the feature map at layer l. Then, we define the style of I as the collection of first-order and second order statistics $\{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L\}$ of the

feature maps, defined as

$$\boldsymbol{\mu}_l = \frac{1}{m_l}\sum_{j=1}^{m_l}\mathbf{F}_l[j] \;\in \mathbb{R}^{p_l} \quad \text{and}$$
$$\boldsymbol{\Sigma}_l = \frac{1}{m_l}\sum_{j=1}^{m_l}(\mathbf{F}_l[j]-\boldsymbol{\mu}_l)(\mathbf{F}_l[j]-\boldsymbol{\mu}_l)^\top \;\in \mathbb{R}^{p_l\times p_l}, \tag{3.1}$$

where $\mathbf{F}_l[j]$ represents the column in $\mathbb{R}^{p_l}$ that carries the activations at position j in the feature map $\mathbf{F}_l$. A style descriptor is then defined as the concatenation of all parameters from the collection $\{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L\}$, normalized by the number of parameters at each layer — that is, $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ are divided by p_l and p_l^2 respectively. This normalization is a slight departure from [44] and [18], but was found to be empirically useful for preventing layers with more output channels to be over-represented. The resulting vector is very high-dimensional, but it contains key information for the description of artistic style. We then apply a singular value decomposition on the style description from the paintings collection to reduce the dimension to 4 096, which empirically keeps more than 99% of the variance in our experiments.

3.1.2 Latent neural network embeddings

As previously mentioned, the method and style description of [18] forms the basis of many state of the art style manipulation methods. This includes conditional instance normalization [12], which represents each style as the parameters of affine transformations, while the rest of the transfer network weights is shared between multiple styles. In [21] this method is extended to handle arbitrary styles, by training two fully connected layers on top of a pretrained Inception-v3 network [67] to *predict* the instance normalization parameters. The first of these fully connected layers is deliberately chosen to produce a 100-dimensional bottleneck, serving as a compact representation of style.

As shown in [21], this 100-dimensional representation turns out to be smooth enough to use it for further analysis of collections of artworks, even though this latent representation remains hard to interpret directly. Therefore, we consider analyzing such a style information by using archetypal analysis as an alternative to the style description based on covariance matrices presented above, in order to show that the conclusions about style decompositions we obtain are generic enough to accomodate several style descriptions. It should be noted though that the loss function used in [21] (as the one in [12]) is still based on the method and loss function of [18], albeit replacing the VGG-19 architecture by VGG-16. It is thus not completely unrelated, but sufficiently distinct and different in structure to make the comparison an interesting one.

With two style decriptions at hand, we can now formulate an optimization problem as described in Formula 2.21 to learn an interpretable representation of style from large collections of artworks, and subsequently perform style decomposition on arbitrary images. To reiterate, the optimization is of the form

$$\min_{\substack{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_n\in\Delta_k\\ \boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_k\in\Delta_n}} \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i-\mathbf{Z}\boldsymbol{\alpha}_i\|^2 \quad \text{s.t.} \quad \mathbf{z}_j=\mathbf{X}\boldsymbol{\beta}_j \text{ for all } j=1,\dots,k, \tag{3.2}$$

and we will take an interest in all of its components:

- The $\mathbf{x}_i$ are the artworks of the dataset (or rather: their style representations) and thus the actual subject of our interest.
- $\mathbf{Z}$ are the archetypes of style (or rather: their style representations) which we will identify during analysis.
- The $\boldsymbol{\beta}$ encode the contribution of styles to the different archetypes.
- The $\boldsymbol{\alpha}$ in turn describe how an artwork's style can be interpreted as a combination of archetypal styles.

We will seek to visualize the latter three, and analyze all four of these.

We apply archetypal analysis independently to the two style descriptions discussed in Sections 3.1.1 and 3.1.2, respectively. That is, each experiment uses either the 4 096-dimensional style vectors obtained from feature statistics, or the 100-dimensional bottleneck features computed by the Inception-v3 model provided by the authors of [21]. We typically learn between $k = 32$ to $k = 256$ archetypes. Each artwork's style can then be represented by a sparse low-dimensional code $\boldsymbol{\alpha}$ in Δ_k, and each archetype is itself associated to only a few input artworks, with the weigts of the contributing styles encoded in $\boldsymbol{\beta} \in \Delta_n$. The sparsity of *both* $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is crucial for their interpretation as we will see the experimental section. Given a fixed set of archetypes $\mathbf{Z}$, we may also quantify the presence of archetypal styles in a new image I by solving the convex optimization problem

$$\boldsymbol{\alpha}^\star \in \operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\Delta_k} \|\mathbf{x}-\mathbf{Z}\boldsymbol{\alpha}\|^2, \tag{3.3}$$

where $\mathbf{x}$ is one of the high-dimensional input style representations described above. Encoding an image style into a sparse vector $\boldsymbol{\alpha}$ allows us to obtain interesting interpretations in terms of the presence and quantification of archetypal styles in the input image. Next, we show how to manipulate the archetypal decomposition by modifying the feature transform of [44].

3.2 Archetypal Style Manipulation

Using the coefficients $(\boldsymbol{\alpha}_i)_{i=1,\dots,n}$ and $(\boldsymbol{\beta}_j)_{j=1,\dots,k}$ described above, we now want to manipulate the style of a given image, natural or artistic. To do so, we extend the universal style transfer method from [44], since it is fast and simple, while providing visually pleasant results.

In the following, we present a modification of the approach of [44] which allows us to better preserve the content details of the original images, before presenting how to use this framework for archetypal style manipulation.

3.2.1 A variant of universal style transfer

As described in Section 2.1.3, style transfer with the Whitening and Coloring Transform (WCT) uses a series of encoder / decoder pairs $(e_l, d_l), l = 1, \dots, L$, with the stylization output at one layer being used as the content image for the next. That is, given an image $\hat{I}_{l-1}$ that is the result of stylization at the preceding layer $l-1$ (with $\hat{I}_0 = I^c$), the image $\hat{I}_l$ is computed at layer l. To do so, we propose the following update, which differs slightly from [44] for a reason we will detail below:

$$\hat{I}_l = d_l\Big(\ \gamma\Big(\delta C_l^s(e_l(\hat{I}_{l-1})) + (1-\delta)C_l^s(e_l(I^c))\Big) + (1-\gamma)e_l(I^c)\ \Big), \tag{3.4}$$

where $\gamma \in [0,1]$ controls the amount of stylization since $e_l(I^c)$ corresponds to the l-th feature map of the original content image. The parameter δ in $(0,1)$ controls how much one should trust the current stylized image $\hat{I}_{l-1}$ in terms of content information before stylization at layer l. Intuitively,

(a) $d_l(C_l^s(e_l(\hat{I}_{l-1})))$ can be interpreted as a refinement of the stylized image computed at layer $l-1$ transferring the mean and covariance structure of the image style at layer l, while

(b) $d_l(C_l^s(e_l(I^c)))$ can be seen as a stylization of the content image by looking at the correlation/mean structure of the style at layer l exclusively, regardless of the structure at the preceding stylization steps.

While $\hat{I}_{l-1}$ takes the style structure of the preceding stylization target layers into account, it may also have lost a significant amount of content information, in part due to the fact that the decoders d_l do not perfectly invert the encoders and do not correctly recover fine details. Obviously, this effect will be more pronounced the more layers the preceding encoders and decoders have. For this reason, being able to make a trade-off between (a) and (b) to explicitly use the original content image I^c at each layer is important.

In contrast, the update of [44] involves a single parameter γ and is of the form

$$\hat{I}_l = d_l\Big(\gamma C_l^s(e_l(\hat{I}_{l-1})) + (1-\gamma)e_l(\hat{I}_{l-1})\Big). \tag{3.5}$$

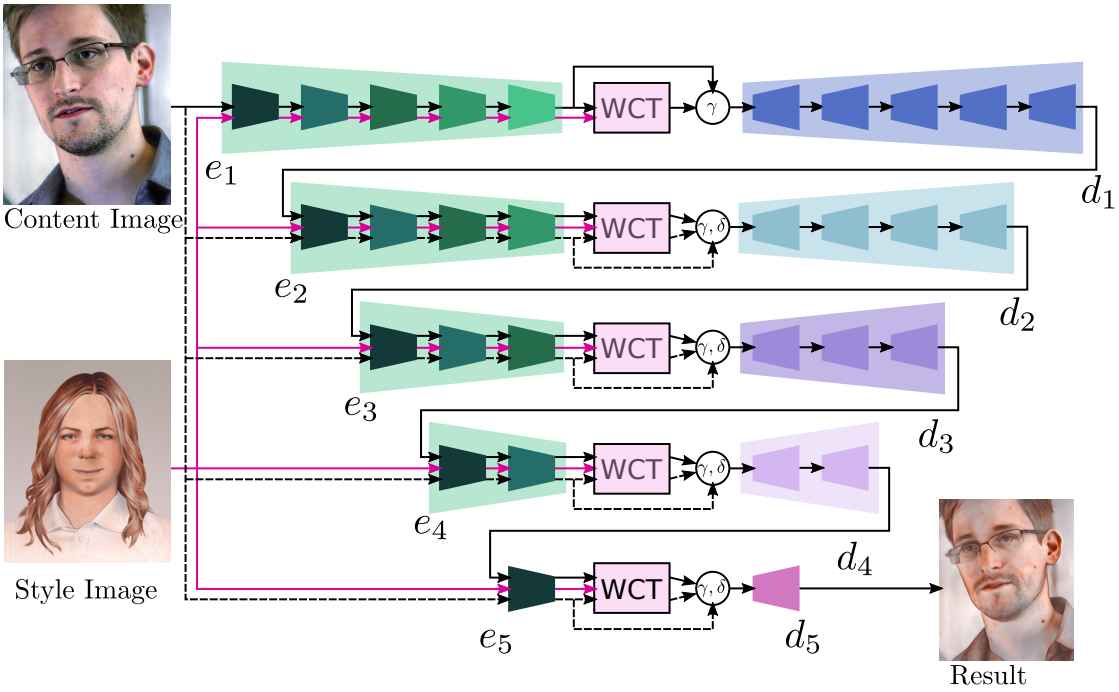


Figure 3.2 – The proposed variant of universal style transfer. We allow for a trade-off between stylization and preservation of detail by making use of the original content image features at every step of the stylization (dashed lines) and by allowing for the use of both stylized and unstylized features thereof by the decoder. The difference to the original method used by [44] can easily be seen when comparing the first encoder/decoder pair (top) with the following pairs.

Notice that here the original image I^c is used only once at the very beginning of the process, and that details that have been lost at layer $l-1$ thus have no chance to be recovered at layer l. Figure 3.2 shows the modified procedure in comparison to Figure 2.4. Obviously this change does not concern the first encoder/decoder pair, since they operate on the orinial content image already. In the experimental section we take a look at the impact of this change on the stylization results. Whenever one is not looking for a fully stylized image — that is, $\gamma < 1$ in (3.4) and (3.5) — content details can be much better preserved with our approach.

3.2.2 Archetypal style manipulation

We now aim to analyze styles and change them in a controllable manner based on styles present in a large collection of images rather than on a single image. To this end, we use the archetypal style analysis procedure described in Section 3.1. Given now an image I, its style, originally represented by a collection of statistics

$\{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L\}$, is approximated by a convex combination of archetypes $[\mathbf{z}_1, \ldots, \mathbf{z}_k]$, where archetype $\mathbf{z}_j$ can also be seen as the concatenation of statistics $\{\boldsymbol{\mu}_1^j, \boldsymbol{\Sigma}_1^j, \ldots, \boldsymbol{\mu}_L^j, \boldsymbol{\Sigma}_L^j\}$. Indeed, $\mathbf{z}_j$ is associated to a sparse code $\boldsymbol{\beta}_j$ in Δ_n, where n is the number of training images—allowing us to define for archetype j and layer l

$$\boldsymbol{\mu}_l^j = \sum_{i=1}^n \boldsymbol{\beta}_j[i]\boldsymbol{\mu}_l^{(i)} \quad \text{and} \quad \boldsymbol{\Sigma}_l^j = \sum_{i=1}^n \boldsymbol{\beta}_j[i]\boldsymbol{\Sigma}_l^{(i)},$$

where $\boldsymbol{\mu}_l^{(i)}$ and $\boldsymbol{\Sigma}_l^{(i)}$ are the mean and covariance matrices of training image i at layer l. As a convex combination of covariance matrices, $\boldsymbol{\Sigma}_l^j$ is positive semi-definite and can be also interpreted as a valid covariance matrix, which may then be used for a coloring operation producing an "archetypal" style.

Given now a sparse code $\boldsymbol{\alpha}$ in Δ_k, a new style $\{\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\mu}}_L, \hat{\boldsymbol{\Sigma}}_L\}$ can be obtained by considering the convex combination of archetypes:

$$\hat{\boldsymbol{\mu}}_l = \sum_{j=1}^k \boldsymbol{\alpha}[j]\boldsymbol{\mu}_l^j \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_l = \sum_{j=1}^k \boldsymbol{\alpha}[j]\boldsymbol{\Sigma}_l^j.$$

Then, the collection of means and covariances $\{\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\mu}}_L, \hat{\boldsymbol{\Sigma}}_L\}$ may be used to define a coloring operation.

Three practical cases come to mind:

(i) $\boldsymbol{\alpha}$ may be a canonical vector that selects a single archetype;

(ii) the vector $\boldsymbol{\alpha}$ may be any convex combination of archetypes for archetypal style interpolation;

(iii) $\boldsymbol{\alpha}$ may be a modification of an existing archetypal decomposition to enhance a style already present in an input image I — that is, $\boldsymbol{\alpha}$ is a variation of $\boldsymbol{\alpha}^\star$ defined in (3.3).

3.3 Experiments

In this section, we present our experimental results on datasets described below. Our implementation uses PyTorch [55] and relies in part on an open-source implementation of universal style transfer [1]. Archetypal analysis is performed using the SPAMS software package [7, 50], and the singular value decomposition is performed by scikit-learn [56]. Further examples can be found on the project website at http://pascal.inrialpes.fr/data2/archetypal_style. In Section 3.3.1, we introduce our dataset; in Section 3.3.3, we show some visualizations obtained using archetypal analysis; Section 3.3.4 is devoted to style manipulation of paintings and Section 3.3.5 exploits meta-data showing that archetypes are able, to some extent, to automatically capture art historical concepts. Unless specified, the

1. https://github.com/black-puppydog/PytorchWCT

style description we use is that of Section 3.1.1, based on feature maps' means and covariances. Section 3.3.6 shows some results when the style representation described in Section 3.1.2 is used instead.

3.3.1 Dataset based on WikiArt

We use a collection of artworks catalogued by the Wikiart [2] project. It consists of 141,910 entries, each being one artwork for which a photo and some metadata is provided. After exclusion of installations, architecture, and some other categories and genres that are not amenable to artistic style transfer such as sketches and studies, we are left with 117,083 paintings and drawings. Most of these have artist and completion year annotations. The dataset also includes information about the artists, like the institutions they worked at, what art movements they were associated with, as well as dominant styles and genres throughout their work. We apply our analysis both to the full dataset, computing $p = 256$ and $p = 64$ archetypes, as well as to subsets of the dataset, for which we only compute $p = 32$ archetypes each. Specifically, we take a separate look at the work of Pablo Picasso, Vincent van Gogh and Salvador Dalí as artists well represented in the dataset, as well as a *groups* of artists, to highlight some of the properties of archetypal style analysis. Below, we discuss the various subsets we consider for our analysis.

Pablo Picasso

The total size of Picasso's oeuvre is estimated to be around 50,000 according to [35]. Of these, only 1,885 are paintings though. Since we are only interested in paintings, our subset of 1,065 (dated between 1890 and 1972) can be expected to give reasonable results. Picasso's work is often grouped into periods, and the Wikiart annotations contain seven such periods. They are, in chronologic order: the early years, blue period, rose period, African period, cubist period (often divided into analytic and synthetic cubism), neoclassicist and surrealist period, and the later years. Most of the paintings in the dataset (all except 46) have a "period" annotation, which we will use to examine our results, but *not* for obtaining them.

Vincent van Gogh

Based on the WikiArt metadata, we exclude a number of works not amenable to artistic style transfer such as sketches and studies. The collection counts 1,154 paintings and drawings in total, with the dates of their creation ranging from 1858 to 1926. Nevertheless, since most of the paintings from this subset are concentrated around 1890, with no metadata describing particular styles that could have evolved

2. https://wikiart.org/

during his career, we will use this subset to perform archetypal analysis, but we will not study the link between archetypes and metadata.

Salvador Dalí

With 1,160 works, the Wikiart collection holds a significant part of Dalí's work. After discarding unsuitable items, we are left with 1,033 works dated between 1913 and 1983. While Dalí's biography does not follow periods as strict as Picasso's, the online catalogue raisonné [3] groups his works into the periods 1910–1929, 1930–1939, 1940–1951, 1952–1964, and 1965–1983. Again, we make use of this metadata only to interpret our results *after* analysis.

Venetian school

We also analyze the styles of a group of artists, namely those of the "Venetian school". The Venetian school is the name given by art historians to a group of artists active in 15th to 18th century Venice that had a lasting influence on western painting. Again, the Wikiart dataset is far from complete; the discussion in [78] lists close to 50 distinct artists of the Venetian school in the collection of the Metropolitan Museum of Art in New York alone, while the Wikiart collection contains only 16 artists with annotations linking them to this group. This small collection of 1,716 paintings, ranging from 1430 to 1789, will nevertheless serve as an example of a group of paintings. These paintings share a chronology and many aspects of style, yet vary as a function of their creators and the time of their creation.

Four well-known artists

Similar to the Venetian school, we select a subset of paintings produced by four of the painters with the most artworks in the Wikiart collection: Claude Monet (1,339 paintings), Camille Pissarro: (881), Pierre-Auguste Renoir (1,334), and Henri Matisse (911). These painters and their styles are closely related, due not only to their being active around the same time, but also due to their social and artistic interactions. Notably though, while Monet, Pissarro, and Renoir had very strong personal contacts, Matisse was born about three decades after Renoir and Monet (four after Pissarro) and so his relationship to the other three is significantly less direct. This collection contains a total of 4,465 paintings.

3. http://www.salvador-dali.org/en/artwork/catalogue-raisonne/

3.3.2 Visualizing the archetypes of a collection

The easiest way to visualize an archetype is to display the paintings that most strongly contribute to its decomposition. That is, we show the paintings associated to the largest non-zero coefficients in the vector $\boldsymbol{\beta}_j$ for archetype j. This allows reasoning about the contributions of individual images' style to the archetype. Figures 3.3 and 3.4 show some examples of archetype visualizations for the full dataset and for the Picasso subset. Visit the project website for the full set of archetypes. The strongest contributions usually exhibit a common characteristic like stroke style or choice of colors. Smaller contributions are often more difficult to interpret These examples illustrate some of the properties of the style representations: in Figure 3.4, archetype 8 summarizes the blue period from Picasso's life using different sets of paintings; later, we will show by using meta-data that the analysis of Picasso's work yields certain archetypes that are consistent with the canonical periods of the artist's life. Similar relationships can be seen in Figure 3.3 when analyzing the full Wikiart dataset.

Not all archetypes are meaningful, or interpretable; there will usually be some that encode properties that are not of interest to us. There are however two failure modes that are typical examples of bad choice of the hyperparameter k. One example of this in which an archetype is comprised of only one artwork is shown in Figure 3.5. While not *necessarily* a sign of bad hyperparameter choice — a very well-positioned data point *can* in principle be useful as an archetype in its own right — these trivial archetypes are usually a sign that the number of archetypes k has been set too high.

The opposite case is setting k to a number that is too low. In this case, the archetypes are forced to cover too much space, subordinating the sparsity loss to the approximation loss term. This is demonstrated with one archetype in Figure 3.6.

In this case the strongest three components only explain about 28% of the archetype. This makes it hard to interpret the archetype. Further examples of this may be seen in some of the archetypes shown on the project website.

Finally, we show in Figure 3.7 an archetype from the analysis of the Venetian school subset that is only composed of portraits, with the strongest contributions being paintings by Giorgione, Tintoretto, and Paolo Veronese. This highlights a limit of style analysis: most methods try (often explicitly) to *disentangle* style from content. In many cases however, disentangling style and content is simply not possible. The example in Figure 3.7 illustrates this: artists of the Venetian school worked in a very different setting from today's artists. The possibility for a painter to develop its own style was very limited in western art for a long time, which meant that they painted portraits and other scenes as demanded by their clients, in the style demanded by the clients. Therefore, a complete description of style for the Venetian school necessarily makes reference to the objects they depict.

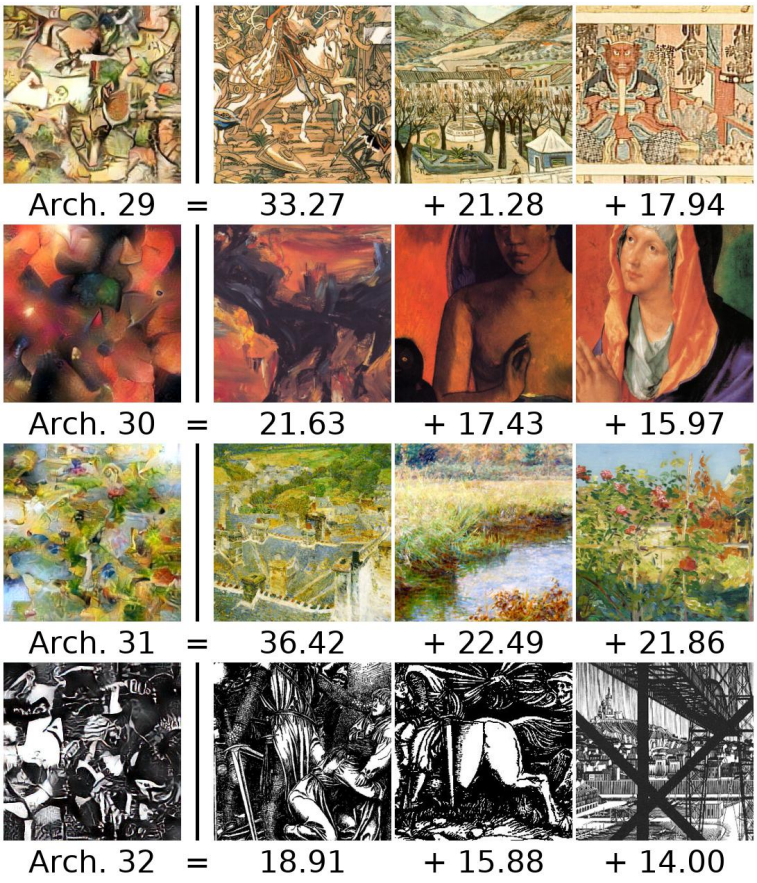


Figure 3.3 – Four archetypes obtained from the full collection when learning $p = 64$ archetypes. The full collection is available on the project website.

In some situations it is necessary to represent an archetype as *one* visualization. For these situations, another way consists in synthesizing a single texture image per archetype that represents its style. As described in [44], this can be done by using a style representation to repeatedly stylize an image filled with random noise, taking the result of the stylization as the content image for the next iteration. After few repetitions of the procedure the image will take on the characteristics corresponding to the chosen style representation

This procedure readily allows for the visualizations of the styles present in a collection and how they relate to one another. Figure 3.8 shows t-SNE [48] embeddings in two dimensions for 256 archetypes computed on the Wikiart collection. Each archetype is represented as one texture, allowing for a concise and intuitive overview of styles in the collection. For example Renaissance and Baroque styles are grouped together at the left of the plot, while abstract and intensely-colored styles occupy the top right.

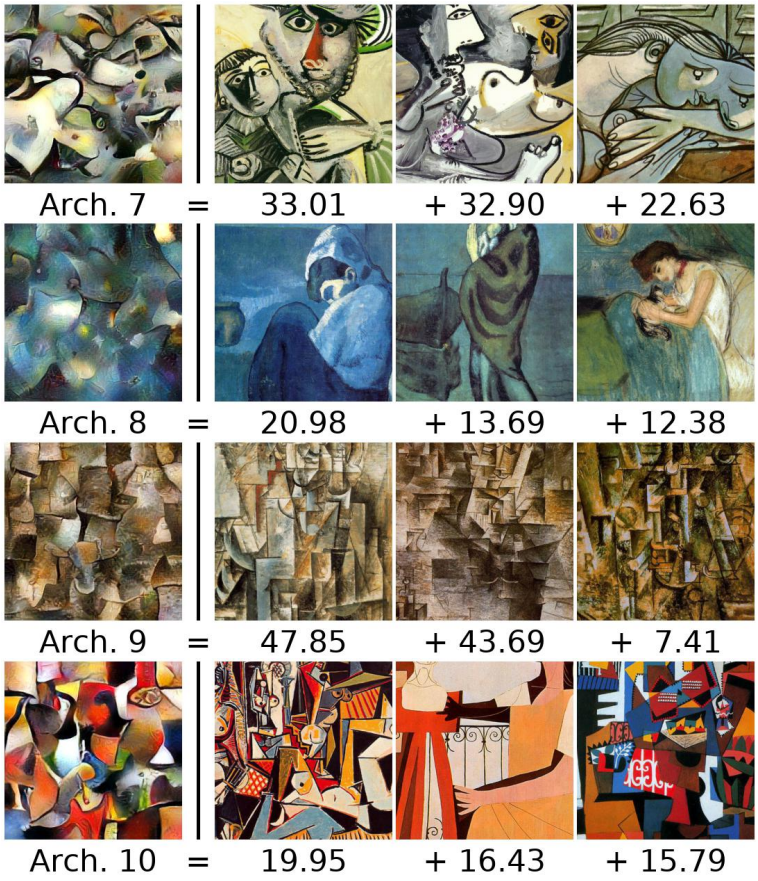


Figure 3.4 – Four archetypes obtained from the Picasso collection when learning $p = 32$ archetypes. The full collection is available in on the project website.

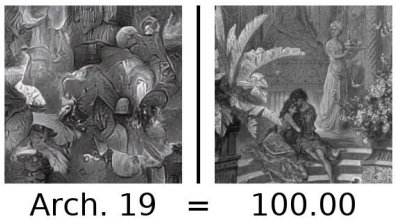


Figure 3.5 – Failure case of a trivial archetype obtained from the full collection when learning $p = 256$ archetypes.

Figure 3.6 – Failure case of a non-sparse archetype obtained from the full collection when learning $p = 64$ archetypes.

Figure 3.7 – Archetype from analysis of the Venetian school.

3.3.3 Picking apart an artwork's style

Similar to showing the decomposition of an archetype into its contributing images, we display in Figures 3.9, 3.10, and 3.11 examples of decompositions of image styles into their contributing archetypes. Given an image, it means that we compute the optimal code $\boldsymbol{\alpha}^\star$ defined in (3.3) and identify which archetypes are selected in the decomposition. We can then in turn display the corresponding archetypes given the visualization approach described in the previous section. Typically, only a few archetypes contribute strongly to the decomposition. Even though often interpretable, the decomposition is sometimes trivial, whenever the image's style is well described by a single archetype. The top row shows a concise visualization of the contributing archetypes as described in Section 3.3.4. For this, too, there are examples on the project website.

3.3.4 Archetypal style manipulation

As noted above, archetypal style analysis cannot only be used for analyzing a collection, but also for manipulating the styles of arbitrary paintings or photographs. Here, we show some examples of this. First though, we study the influence of the parameters γ, δ and make a comparison with the baseline method of [44]. Even though this is an apparently minor modification, it yields significant improvements in terms of preservation of content details in stylized images. Besides, the heuristic $\gamma = \delta$ appears to be visually reasonable in most cases, reducing the number of

Figure 3.8 – t-SNE embeddings of 256 archetypes computed on the Wikiart collection. Each archetype is represented by a synthesized texture. Best seen by zooming on a computer screen.

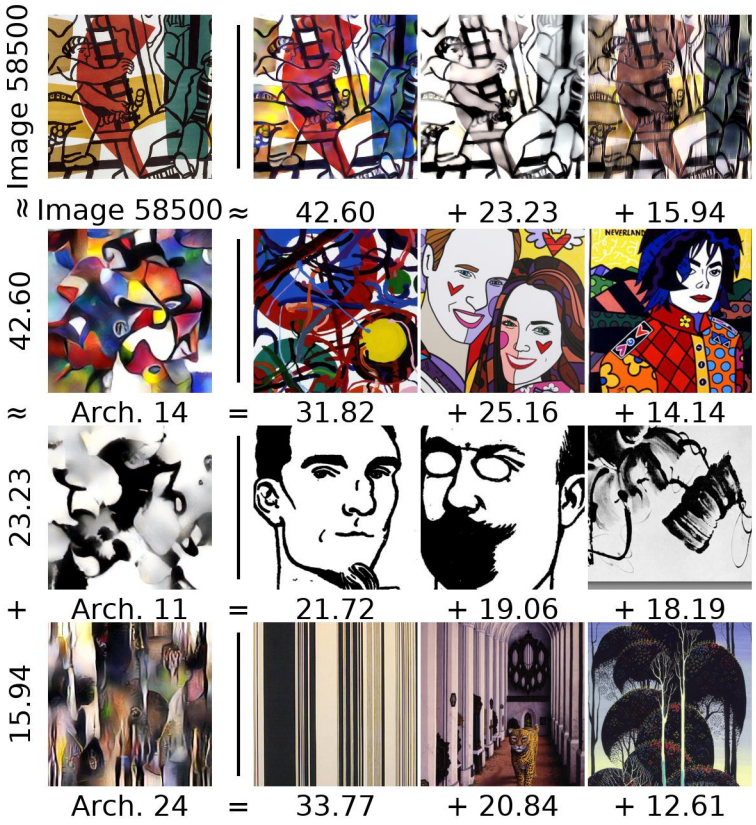


Figure 3.9 – Image decomposition from the full dataset. Each archetype is represented as a stylized image (top), as a texture (side) and as a decomposition into paintings. Note that the bottom archetype seem to encode consecutive vertical bars.

effective parameters back to a single one that controls the amount of stylization. The comparison between our update (3.4) and (3.5) from [44] is illustrated in Figure 3.12, where the goal is to transfer an archetypal style to a Renaissance painting. At $\gamma = \delta = 1$, the approaches are equivalent, resulting in equal outputs. Otherwise however, especially for $\gamma = \delta = 0$, [44] produces strong artifacts. These are not artifacts of stylization, since in this case, no actual stylization occurs. Rather, they are the effect of repeated lossy encoding and decoding, since no decoder can recover information lost in a previous one. More comparisons on other images and illustrations with pairs of parameters $\gamma \neq \delta$, as well as a comparison of the processing workflows, are provided in Appendix A, confirming our conclusions.

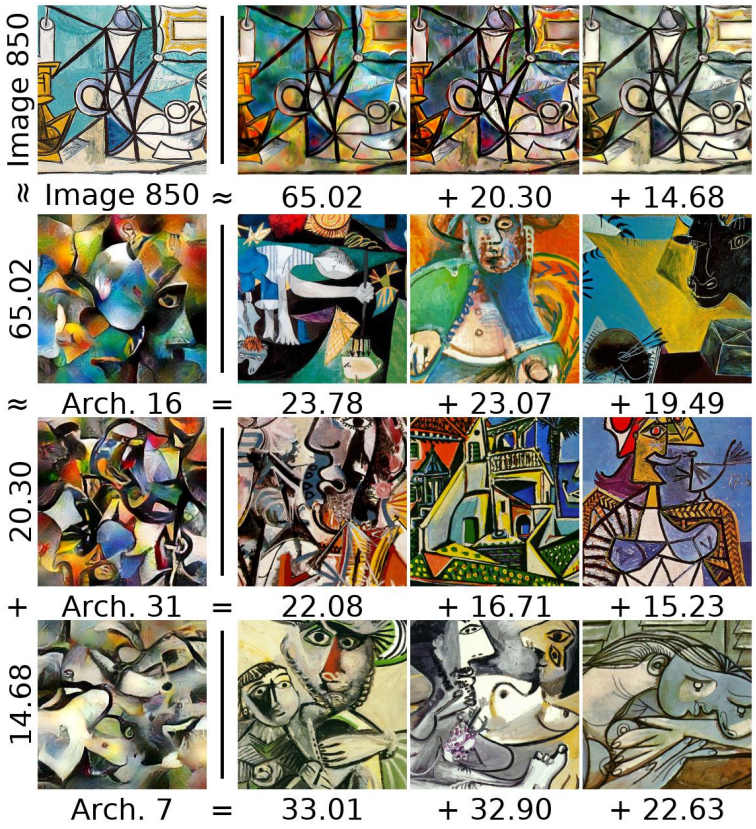


Figure 3.10 – Image decomposition of "An artist" by Picasso (1963) into archetypes computed from the Picasso subset.

Style enhancement

To obtain variations of an input image, the decomposition $\boldsymbol{\alpha}^\star$ of its style can serve as a starting point for stylization. Figure 3.13 shows the results of enhancing the archetypes which an image already exhibits. Intuitively, this can be seen as taking one aspect of the image, and making it stronger with respect to the other ones. In Figure 3.13, while increasing the contributions of the individual archetypes, we also vary $\gamma = \delta$, so that the middle image is very close to the original image ($\gamma = \delta = 0$), while the outer panels put a strong emphasis on the modified styles. These *extreme styles* are also shown in Figures 3.9, 3.10, and 3.11 in the top row. They can serve as a visualization of possible directions one can take when manipulating the style of an image and starting from its original style. As can be seen especially in the panels surrounding the middle, modifying the decomposition coefficients allows gentle movements through the styles.

The leftmost and rightmost panels of Figure 3.13 however show that enhancing the contribution of an archetype can produce significant changes too.

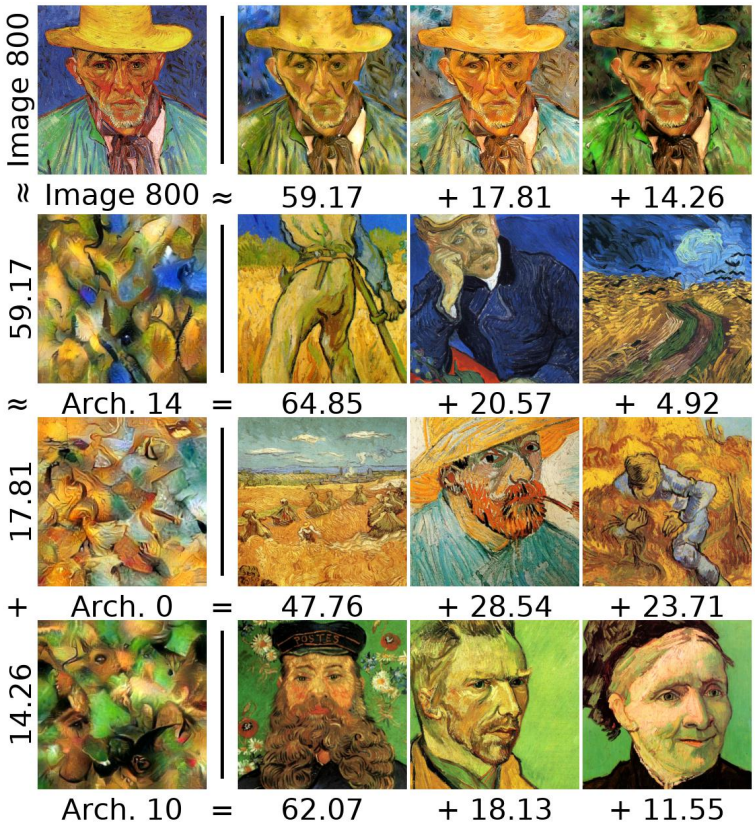


Figure 3.11 – Image decomposition of "Portrait of Patience Escalier, Shepherd in Provence" by van Gogh (1888) into archetypes computed from the van Gogh subset.

Figure 3.12 – Top: stylization with our approach for $\gamma = \delta$, varying the product $\gamma\delta$ from 0 to 1 on an equally-spaced grid. Bottom: results using [44], varying γ. Best seen on a computer screen.

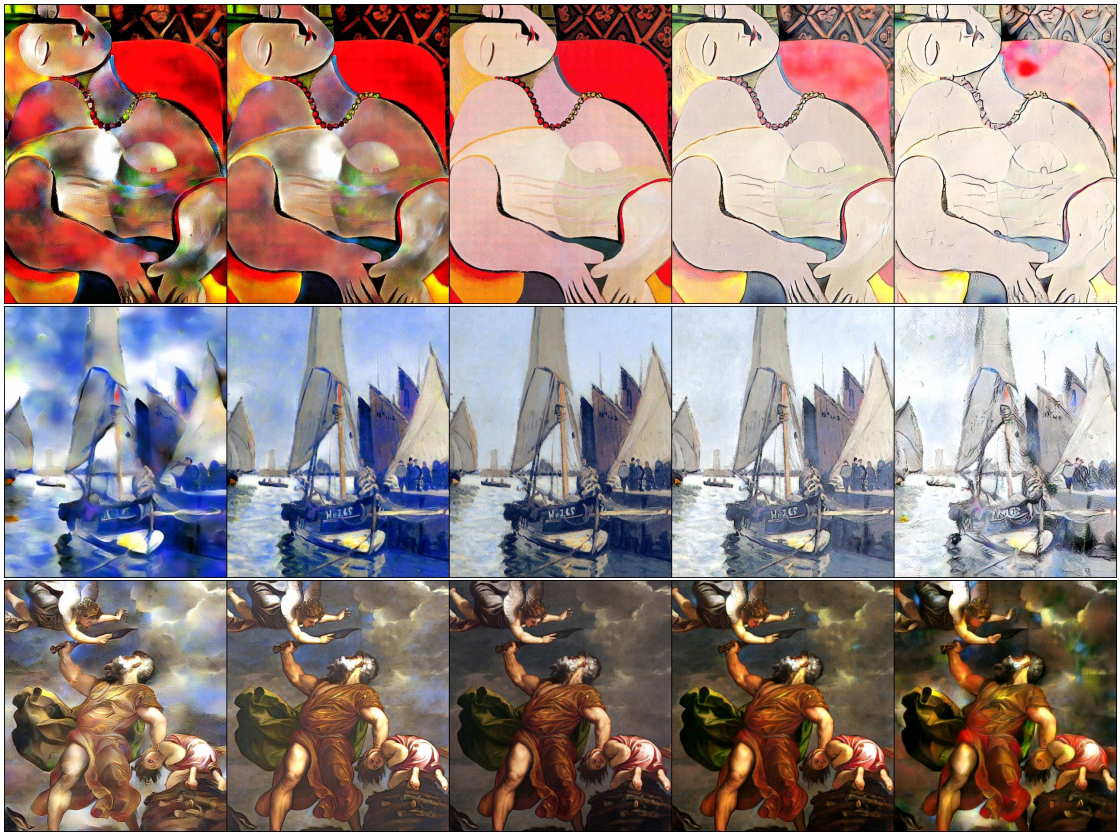


Figure 3.13 – We demonstrate the enhancement of the two most prominent archetypal styles for different artworks. The middle panel shows a near-perfect reconstruction of the original content image in every case and uses parameters $\gamma, \delta = 0$. Then, we increase the relative weight of the strongest component towards the left, and of the second component towards the right. Simultaneously, we increase $\gamma\delta$ from 0 in the middle panel to 0.95 on the outside. From top to bottom: "A Dream" by Pablo Picasso, archetypes computed on the Picasso dataset. "Sailing Boats at Honfleur" by Claude Monet, archetypes computed on the full Wikiart dataset. "Sacrifice of Isaac" by Titian, archetypes computed on the Venetian school dataset.

Free manipulation

Naturally, it is also possible — and sometimes desirable, depending on the user's objective — to manually choose a set of archetypes that are unrelated to the input image, and to then interpolate with convex combinations of these archetypes. This results in images akin to those found in classical artistic style transfer works. In Figure 3.14, we apply for instance combinations of freely chosen archetypes to "Cock and Knife" by Pablo Picasso, and in Figure 3.15, we perform a similar experiment to "The Bitter Drunk" by Adriaen Brouwer. The fact that we are manipulating paintings here is however completely incidental in this setting; Figure 3.16 shows the application of the very same procedure to a well-known natural image.

Other examples are also provided on the project website.

3.3.5 Relationship to art historic concepts

For the "Four Artists" subset, we observe that many archetypes are composed almost exclusively of paintings from a single artist. Figure 3.17 shows some of these. This observation is encouraging since during analysis, we have not made use of any annotations present in the Wikiart dataset; indeed, our method learns the archetypal representation of style in a completely unsupervised fashion.

It is also worth pointing out that while the VGG-19 network that we use for feature extraction (and that is also used for supervision while training the Inception-v3 network) was trained in a fully supervised way, it was trained on the Imagenet dataset, and so was given no supervision related to artistic style. While this representation can be put to use for the exploration of collections and style modification as described above, it is interesting to relate it to concepts of art history. The representations of style that we use are motivated mostly by the pragmatic need for effective texture generation and stylization, and are not explicitly trained to relate to concepts such as artists or the chronology of style development *across* artists. It is however of interest to find representations of style that do capture aspects of art historic concepts. As a first step in this direction, we examine if the representations that our method learns in this completely unsupervised fashion capture any information that humans would use to communicate about art history. To this end, we take a look at the way that archetypes decompose into styles from different groups of paintings, and which paintings' styles they principally contribute to, i.e. the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ computed during archetypal analysis.

Four artists

Following up on the subjective impression gained from Figure 3.17, we look at how the archetypes are composed from paintings of the different artists, and which

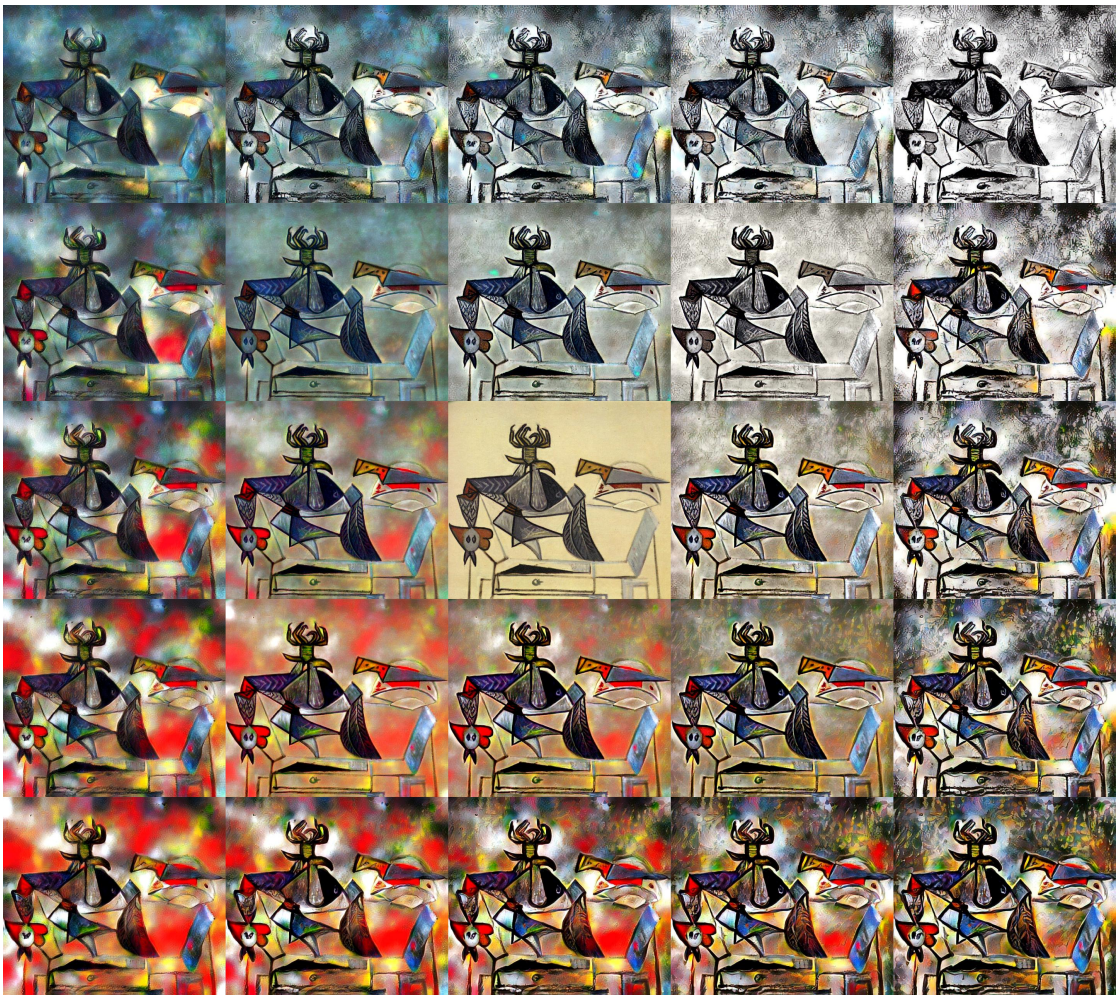


Figure 3.14 – Free archetypal style manipulation of "Cock and Knife" by Pablo Picasso. The middle shows the original picture ($\gamma = \delta = 0$) and the results on the diagonals are stylizations for the four chosen archetypes. As for the style enhancement experiments above, we increase γ and δ from the inner to the outer ring. Archetypes computed on the Picasso dataset.

Figure 3.15 – Free archetypal style manipulation of “The Bitter Drunk” by Adriaen Brouwer. Unlike Figure 3.14, the center image does not correspond to the original image, but to a stylized one obtained by setting the four archetypes to 25% each.

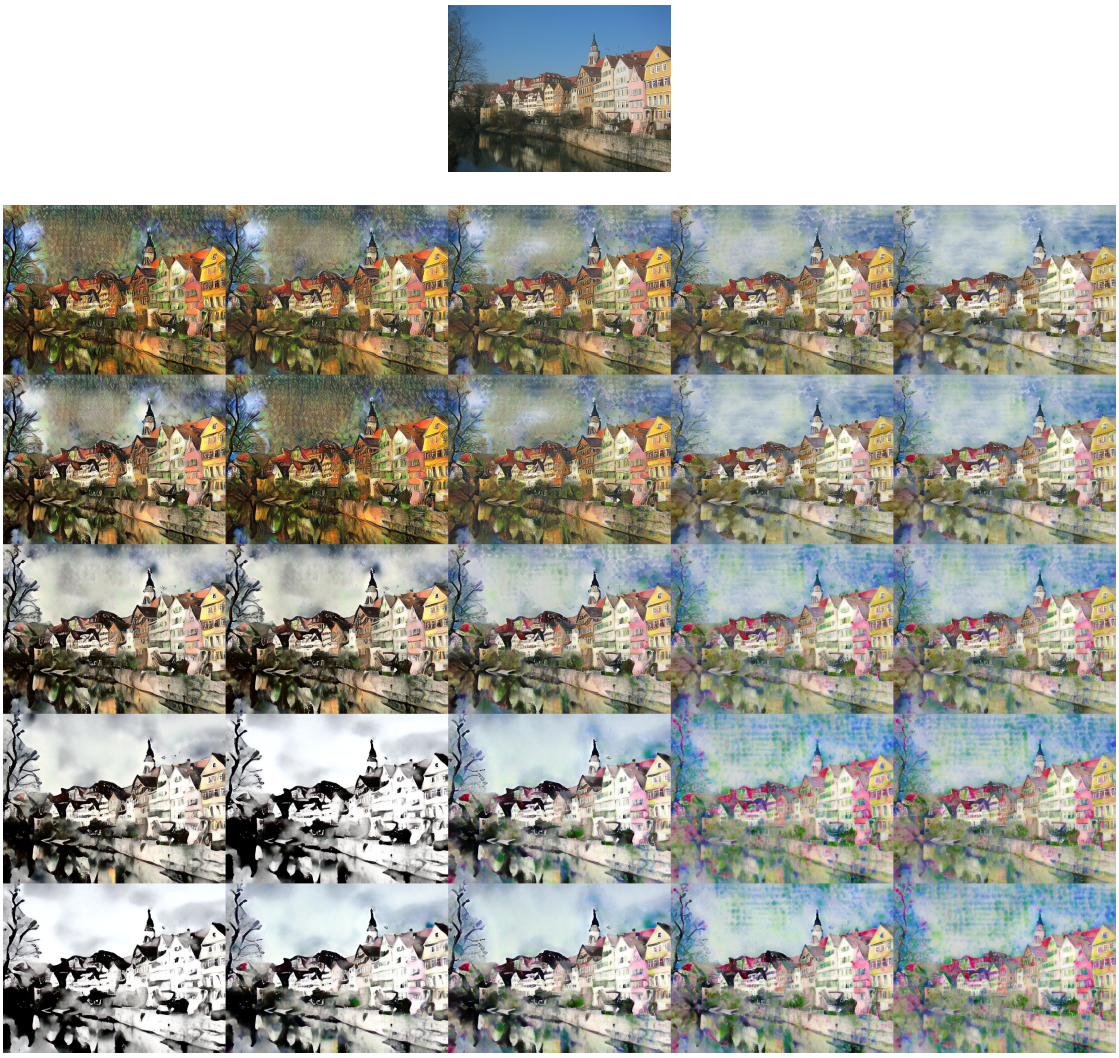


Figure 3.16 – Free stylization of “Tübingen” photo.

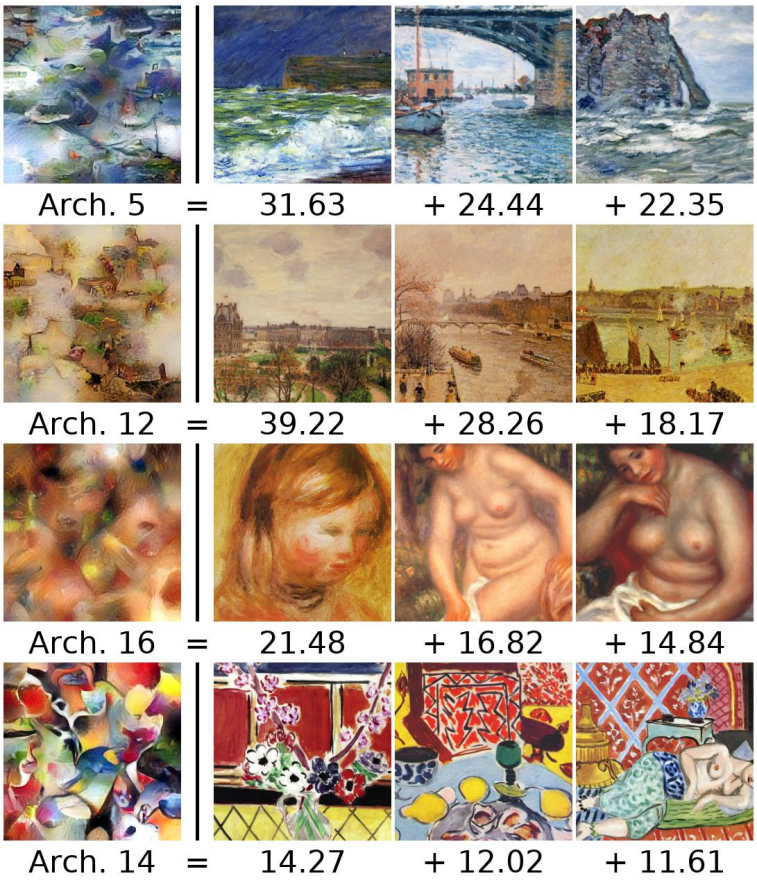


Figure 3.17 – Archetype from analysis of the "Four Artists". Each archetype shows an aspect of one artist's work. From top to bottom: Monet, Pissarro, Renoir, Matisse.

artists' paintings they contribute to.

In the top panel of Figure 3.18, we add up, for each artist and each archetype, how much paintings by that artist contribute to that archetype, and normalize each column to sum to one. We sort the archetypes (columns) by the artist name with the strongest contribution to that archetype, so we get a descending pattern. In the bottom panel we add up, for every artist and every archetype, the contribution of that archetype to that artist's paintings. We sort the archetypes with the same order as in the top figure. We normalize each column to sum to 1, so we cannot assess the relative importance of the archetypes, but we see which artists' paintings an archetype contributes to. Importantly, in the lower panel we discard paintings that contribute significantly ($\beta_{ji} >= 0.1$) to at least one archetype in order to remove trivial associations. These paintings naturally get strong contributions back from the respective archetypes, but those contributions are hardly surprising; if

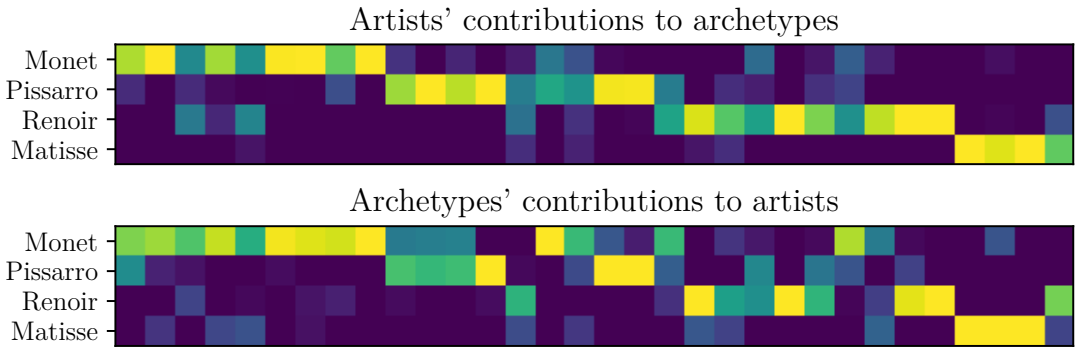


Figure 3.18 – Top: contributions of artists to archetypes. Bottom: contributions of archetypes to artists. Each column is devoted to one archetype.

style descriptor contributes strongly to an archetype, that archetype will end up close to the style descriptor, and will thus yield a low approximation error if used to reconstruct that descriptor.

When manually choosing the color mapping in the figure appropriately, two similar descending patterns become visible in both panels, indicating that archetypes composed from paintings of a certain painter are more likely to also contribute to the other paintings from the same painter.

Venetian School

Whereas the previous subset involves four different artists with relatively different styles, the Venetian school subset is much more challenging since it involves more painters with visually closer artistic styles. Using the same simple visualization as in the previous section, we nevertheless find a similar pattern in the archetypes computed for the Venetian school. Despite the style representation using no information about artists at all, Figure 3.19 shows a similar falling curve pattern in both panels. This shows that there is indeed a positive correlation between the artists that an archetype is comprised of and the artists who's paintings it contributes to.

Biographic development of style

One of the first important books on art history, Giorgio Vasari's "Lives of the Most Excellent Painters, Sculptors, and Architects" [73], introduced the model of analyzing an artist's work along biographical lines. Following along with this established practice of art history, we choose to analyze the works of Pablo Picasso, as an example of an artist with a pronounced development of style throughout his

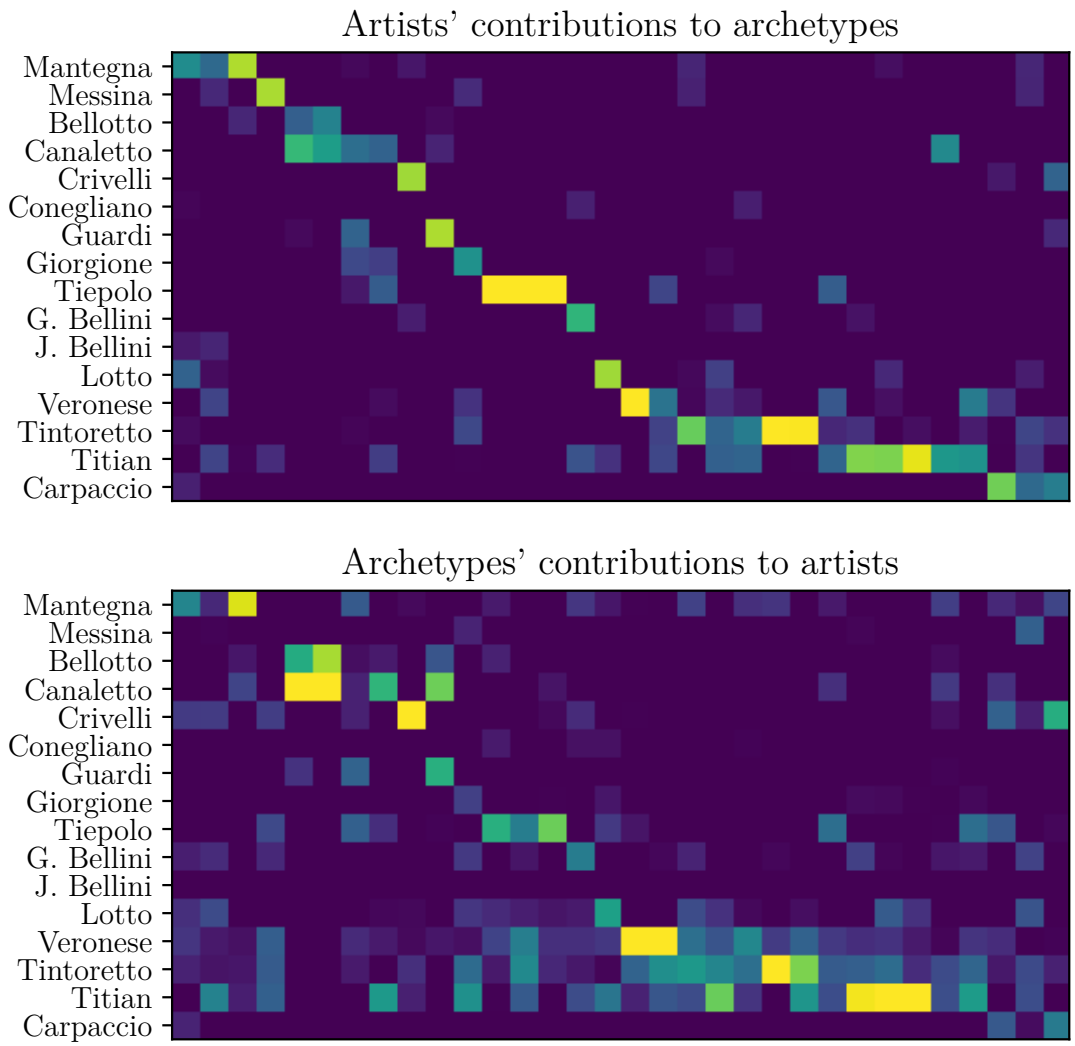


Figure 3.19 – Archetype composition and use for the Venetian school. Top: contributions of artists' paintings to archetypes. Bottom: contributions of archetypes to artists' paintings. Each column is devoted to one archetype.

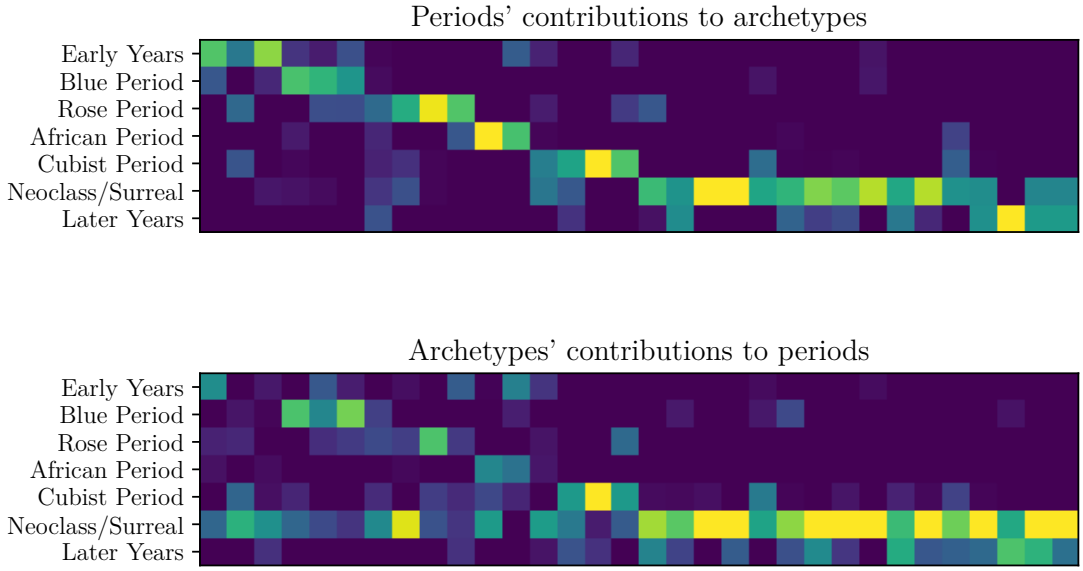


Figure 3.20 – Archetype composition and use for Picasso's work. Top: contributions of paintings from different periods. Bottom: contributions of archetypes to paintings from the different periods. Each column is devoted to one archetype. For this analysis, we exclude the 46 artworks without period annotation.

life. Again, we look at the contributions to and by the archetypes in the form of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, this time grouping the paintings by their assigned periods.

Figure 3.20 shows the resulting visualization, indicating the relation between period label and the contributions of each archetype. Again, we see two similar patterns between, indicating a positive correlation between a period's style contributing to an archetype, and that archetype contributing to other paintings from the same period. While all of the above correlations are weak, they imply that archetypal style may capture some salient differences between styles of paintings that relate to the concept of style as applied in an art historic context.

In Figure 3.21, we perform a similar experiment with Dali's work, whose styles are less discriminative in terms of local image statistics. When organized in periods, the archetypes seem to capture correctly its early work (1910–1929), and most recent ones (1964–1983), but fail to discriminate between the periods 1940–1951 and 1952–1964, showing as well some limitations of our approach, when the style in terms of art historical concepts is strongly related to the global scene organization and content.

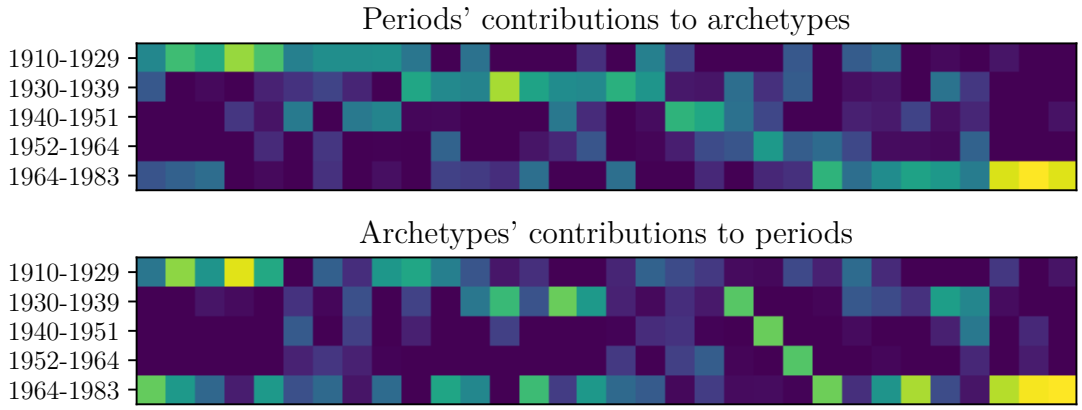


Figure 3.21 – Archetype composition and use for Dali's work. Top: contributions of paintings from different periods. Bottom: contributions of archetypes to periods. Each column is devoted to one archetype.

3.3.6 Evaluation of the Inception network's latent embedding

In Figure 3.22, we display some archetypes learned when using the neural style embedding described in Section 3.1.2 on the Picasso subset. We subjectively find the set of archetypes to be less consistent in the sense that dissimilar paintings in terms of style (again subjectively) are sometimes grouped together. This observation is confirmed in Figure 3.23, where the relation between archetypes and art historical concepts is not as pronounced as in Figure 3.20.

Figure 3.22 – Four archetypes obtained from the Picasso collection when learning $p = 32$ archetypes and using the inception-v3 representation of style described in Section 3.1.2.

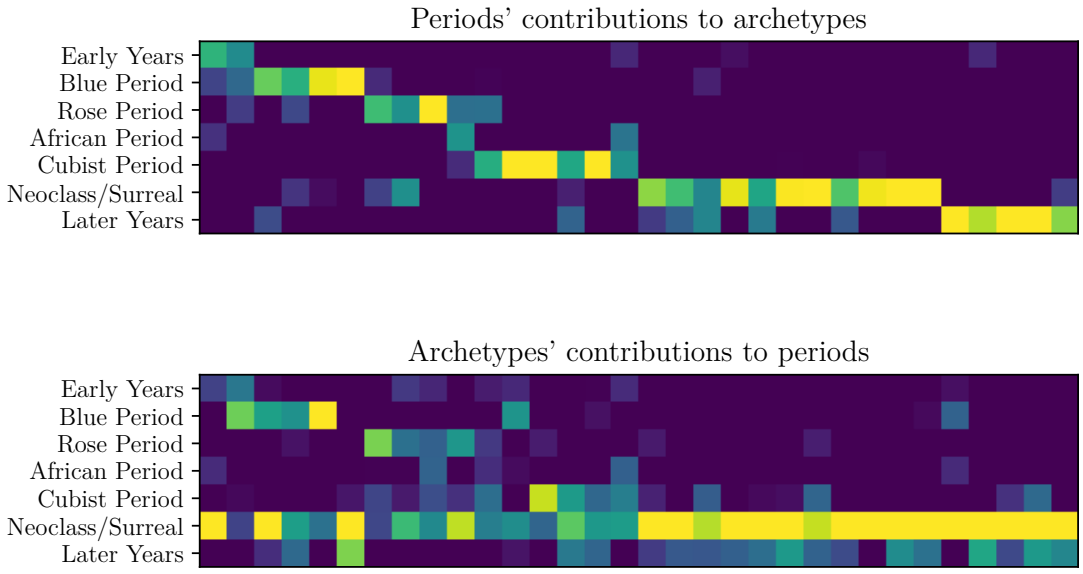


Figure 3.23 – Archetype composition and use for Picasso's work, with archetypes computed on Inception style. Top: contributions of paintings from different periods. Bottom: contributions of archetypes to paintings from the different periods. Each column is devoted to one archetype. For this analysis, we exclude the 46 artworks without period annotation.

3.4 Discussion

In this chapter, we have introduced archetypal style analysis as a means to identify styles in a collection of artworks without supervision, and to use them for the manipulation of artworks and photos. Archetypal analysis admits a dual interpretation which makes it particularly appropriate for the task: on the one hand, archetypes are represented as convex combinations of input image styles and are thus directly interpretable; on the other hand, an image style is approximated by a convex combination of archetypes allowing various kinds of visualizations. Besides, archetypal coefficients may be used to perform style manipulations.

Our method works with different underlying representations of style, and style manipulation can in principle be performed with any method allowing for interpolation between styles. Of course, the quality of the analysis and visualization will always depend on the richness of information encoded in the underlying representation of style. We find that both the representation chosen by Gatys *et al.*, as well as that of Ghiasi *et al.* allow for a fruitful analysis of the datasets we use, despite the fact that both are targeting the task of style *manipulation* first and foremost. The archetypes that are identified on small datasets seem to capture some aspects of art historic relevance, which we evaluate by using meta-data gathered from WikiArt.

Chapter 4

Conclusion

In this dissertation we have presented work on the analysis and manipulation of artistic style using convolutional neural networks. We worked on the description and summarization of artistic style, with a focus on human interpretablility. To this end, we chose to leverage recent developments in neural networks, trained for the classification of natural images, which had previously beeen utilized by others for the purpose of artistic style transfer.

Starting from these dense, high-dimensional representations, we applied archetypal analysis to obtain a much lower-dimensional, sparse representation of style. This *archetypal* representation of style naturally inherits some of the properties of the underlying stylization methods, as well as properties of the analysis methods.

Archetypal analysis — when used with care — produces analysis results that are sparse, a property which, together with the convexity constraints of the formulation, leads to representations that lend themselves to human reasoning. For the case of artistic style, they match quite closely how at least a layperson might communicate about artistic styles. This allows the exploration of large collections of artworks in a visual manner, showing as an analysis result concise summaries of archetypal styles that have been identified therein. Using the symmetric formulation of archetypal analysis, it is also possible to relate a single artwork to these styles, and therefore to artworks inside the collection.

The neural style representations we applied the analysis to were originally used for artistic style transfer. As such, they allow for interpolation of styles and texture generation, both of which can be used for our purposes. Via texture generation, we showed how to summarize the style archetypes in a single image, allowing for much more direct visualizations of the archetypal simplex if necessary. The interpolation

between archetypes allows the archetypal styles to be used for style manipulation after the analysis. This enables intuitive control over the stylization of natural images. It can, however, also be applied to artistic images. Analyzing the style of an artwork, and then making modifications to its archetypal representation, allows making subtle changes in style that mostly preserve the appearance of the result.

Specific to our choice of the Whitening and Coloring Transform (WCT) as the stylization method, we presented a modification that allows for better control over the tradeoff between stylization strength and preservation of detail. This tradeoff only becomes *necessary* when *changing* the style of an existing artwork. It can however also be useful when employing WCT for artistic purposes, applied to natural images.

Finally, to demonstrate some strong and weak points of archetypal style analysis, we applied the analysis to different artists' and groups of artists' works. The resulting analysis underlined some findings from earlier sections — such as the inherent entanglement of content and style in some contexts — and also showed that while the neural representations for style manipulation *do* capture some aspects relating to actual art history as practiced by humans, they leave much to be desired in this regard. This is not exactly surprising; after all, these methods were created with the specific goal of enabling style transfer in mind. It only stands to reason that they won't necessarily excel at other tasks, even if these also relate to artistic styles. What's more: the representations of style commonly used in the style transfer literature often don't even contain information about artistic style learned from data, since they are trained on natural images only.

These two shortcomings both come down to a lack of training specifically with and for data pertaining to artistic style. This does show a path to further research. Incorporating supervision in the form of annotations already available for many artworks should allow the resulting representations to describe artistic style in a much more nuanced way. Indeed, other works in the domain have previously attempted to use supervised problems to learn representations of style that carry more specialized semantic meaning. One attempt to use annotations is [9] which tries to directly make use of artists' mutual *relationships*. More notably though, and with more impressive visual results, [62] makes use of a classifier trained for *artist* (but not *style*) classification. This classifier is then used as an additional, "content-aware" loss term for style transfer training. It seems that in order to become more useful for the purposes of art exploration, maybe even to the point of becoming a tool for scholars, style representations will have to make use of as much information as can be made available to them.

We conducted a preliminary study to gauge the feasibility of such approaches. For this we used the artist, style, and genre labels as classification targets for VGG and VGG-like networks. This study pointed to two practical challenges. First,

the VGG architectures commonly used for style transfer have so many parameters that training them can be challenging even with modern optimizers. Replacing VGG with more modern architectures such as ResNets allows for easier training of the proxy problems. Indeed, an off-the shelf Resnet18, fine-tuned on the WikiArt dataset, trained on a joint classification loss for artist, style, and genre labels, easily achieved about 70% accuracy on the artist and genre tasks and 50% on the style task. However, ResNets yield visually inferior results to the VGG architecture, at least when employing optimization-based style transfer. This can be alleviated to an extent by the use of decorrelated feature spaces in which to do the optimization, as proposed by [53]. But if the goal is to set a new state of the art in neural style transfer, this modification in itself will not be sufficient. Second, the resulting networks when not using VGG-like architectures seemed especially vulnerable to adversarial perturbations of their inputs. In fact, the more weight was placed on deeper layers' loss terms, the stronger those artifacts get. This was already the case for the VGG networks commonly used for neural style transfer, and it got worse when using the deeper ResNet architecture. Consequentially, using deeper networks for optimization-based style transfer yielded no visual improvement over the classic VGG architectures used in the literature. Upon closer inspection, the resulting images showed artifacts commonly seen in the adversarial example literature. This would suggest that robust training methods can be a component in the training of supervised representations of artistic style.

The style *manipulation* aspect of our work, too, should benefit greatly from descriptions that more closely align with human-defined categories of style. Ideally, manipulations might be possible using these categories as targets. Moving a photo or drawing closer to a particular style, and not just closer to a specific *example* of that style, would allow artists to use such manipulations in new and ever-exciting ways.

Appendix A

Further Examples

Contents

Here, we presents a set of additional results, which were not included in the paper for space limitation reasons, as well as experimental material such as the full set of archetypes learned by our approach.

A.1 Influence of γ, δ and comparison with WCT

In this section, we provide additional comparisons between our variant of [44] and the original one. All cases seem to confirm that (i) the heuristic $\gamma = \delta$ is reasonably good in terms of quality of the results, and (ii) our variant is much more accurate than [44] in terms of content preservation as soon as the amount of stylization is less than 100%. The comparison are provided in Figures A.1, A.2, and A.3.

(a) Images produced by our approach when varying δ and γ.

(b) Images produced by our approach when $\gamma = \delta$, jointly increasing these parameters from 0 (left) to 1 (right).

(c) Images produced by the original approach of [44] when changing their stylization parameter.

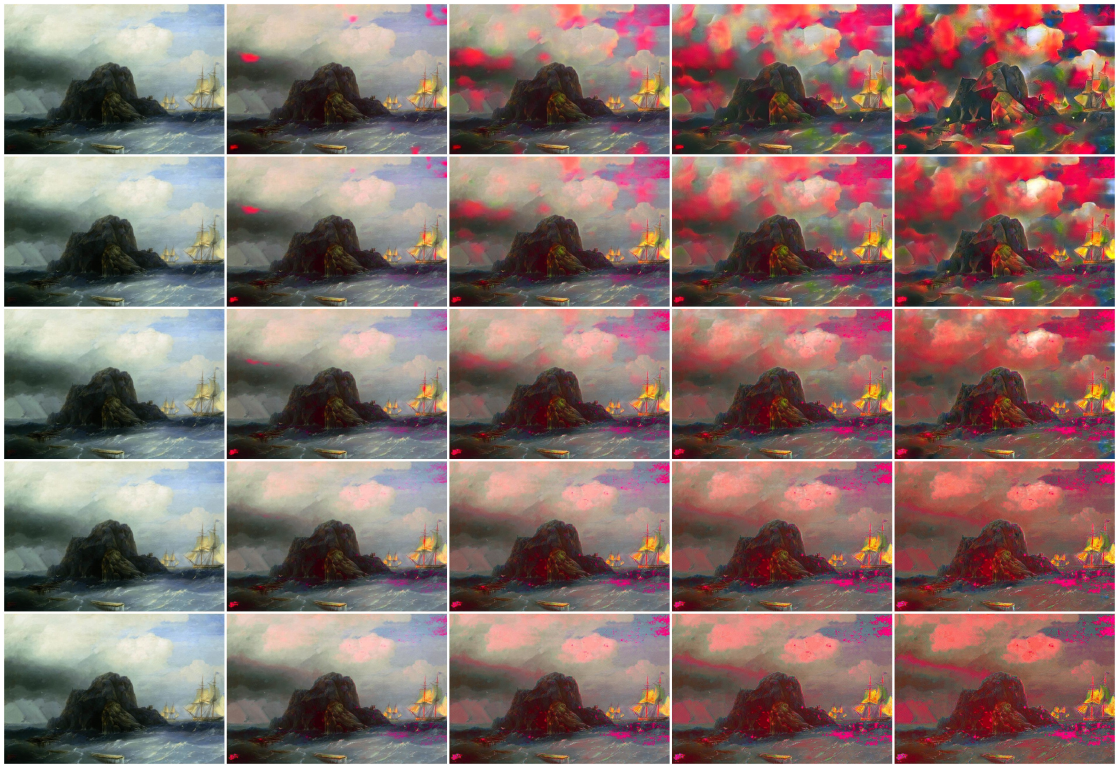
Figure A.1 – Comparison of stylization control between our approach and [44].

(a) Images produced by our approach when varying δ and γ.

(b) Images produced by our approach when $\gamma = \delta$, ranging from 0 (left) to 1 (right).

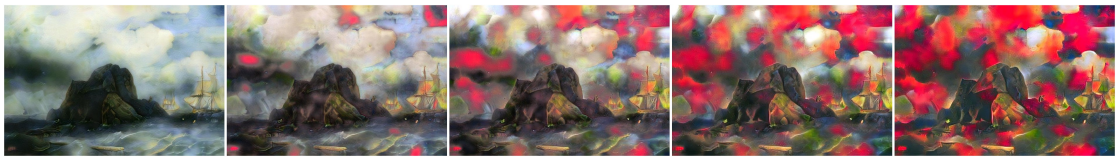
(c) Images produced by the original approach of [44].

Figure A.2 – Comparison of stylization control between our approach and [44].



(a) Images produced by our approach when varying δ and γ.

(b) Images produced by our approach when $\gamma = \delta$, jointly increasing these parameters from 0 (left) to 1 (right).

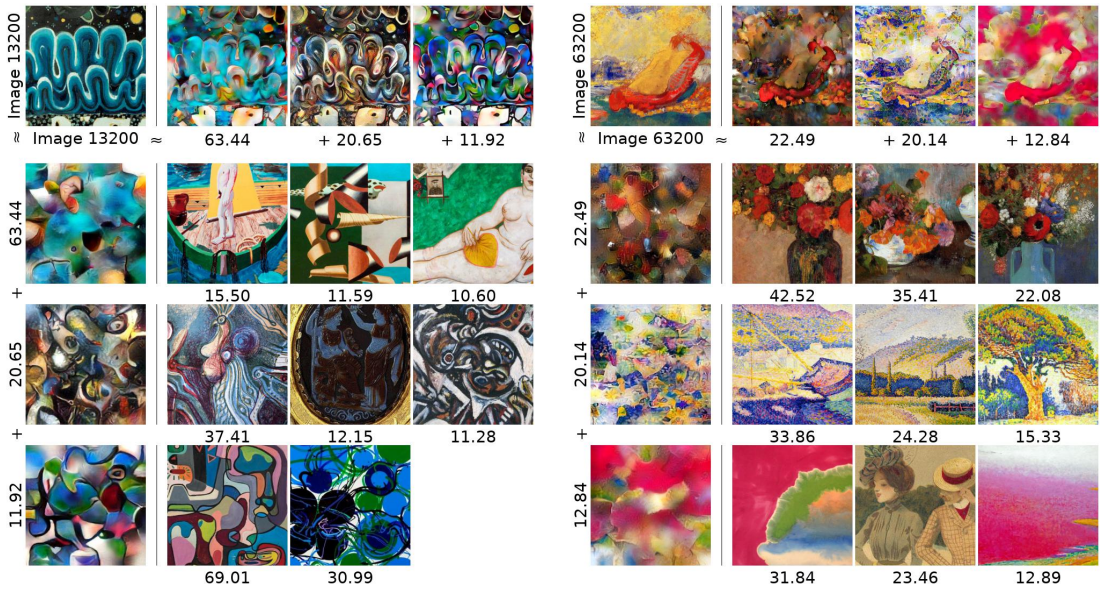


(c) Images produced by the original approach of [44] when changing their stylization parameter.

Figure A.3 – Comparison of stylization control between our approach and [44].

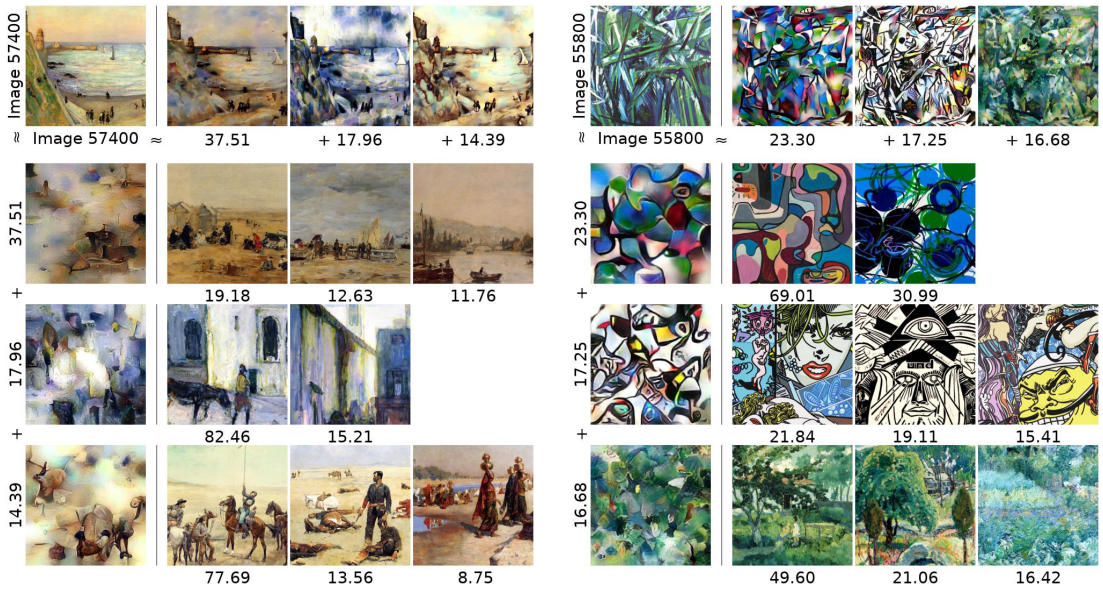
A.2 Examples of Image Decompositions

We show in this section a few additional image decompositions, involving trivial ones, meaningful ones, and failure cases.



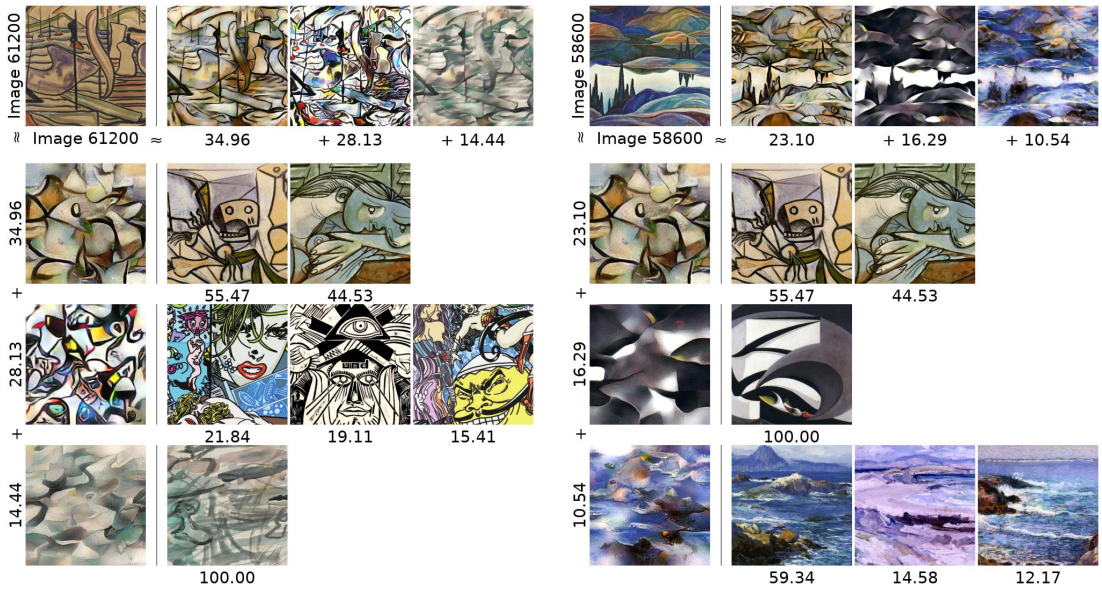
(a) (b) Image decomposition.

Figure A.4 – Image decompositions from the GanGogh collection.



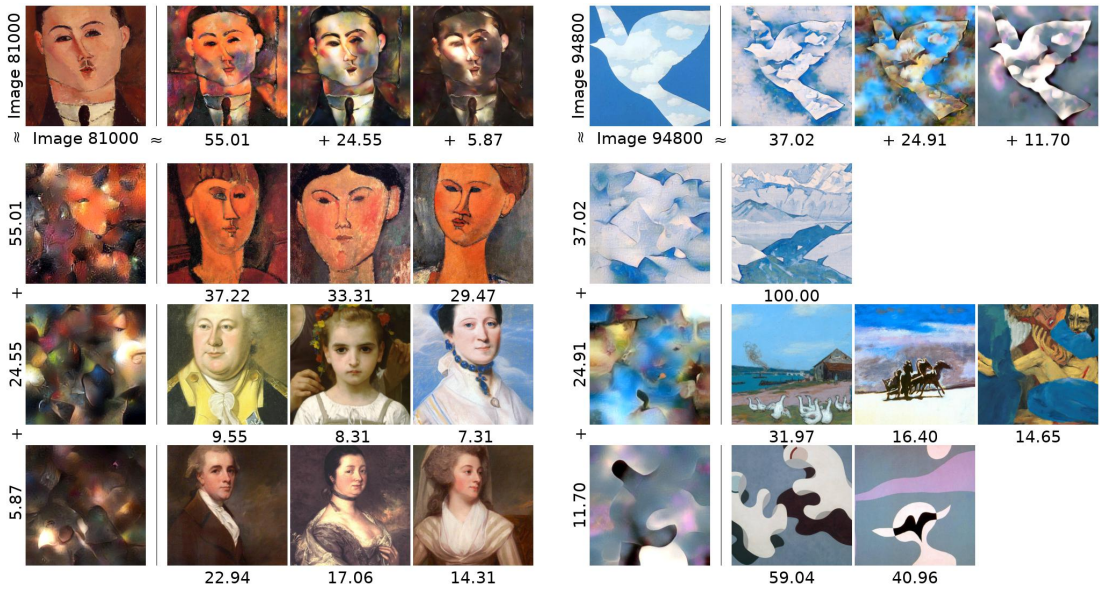
(a) (b) Image decomposition.

Figure A.5 – Image decompositions from the GanGogh collection.



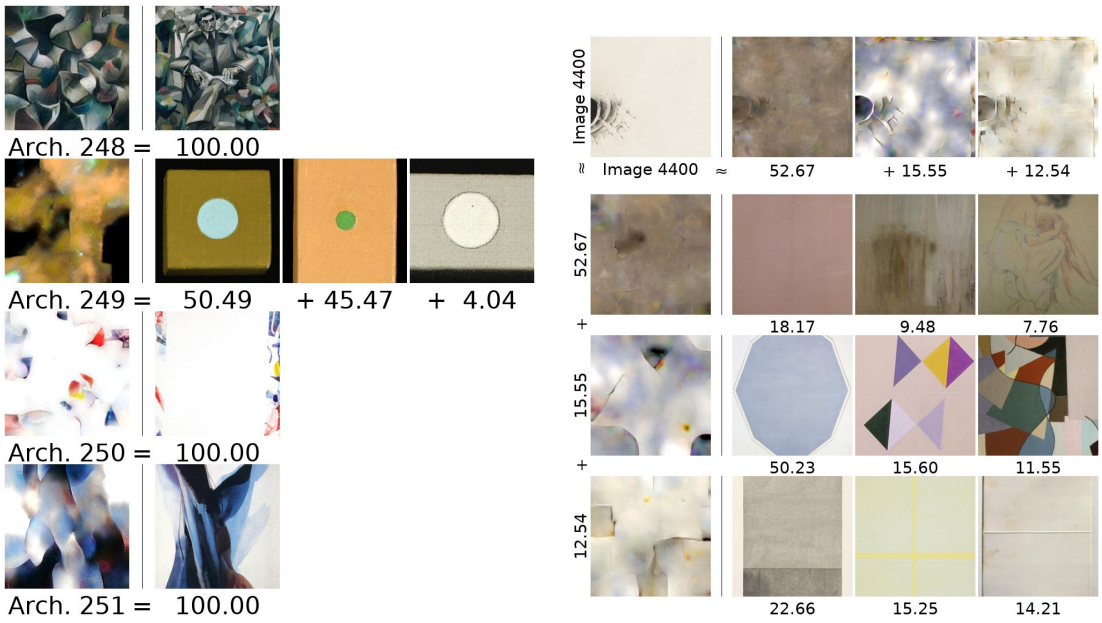
(a) (b) Image decomposition.

Figure A.6 – Image decompositions from the GanGogh collection.



(a) (b) Image decomposition.

Figure A.7 – Image decompositions from the GanGogh collection.



(a) Four archetypal decompositions.

(b) Image decomposition.

Figure A.8 – Failure cases of two archetypal decompositions (a) and image decomposition (b). (a): the second archetype seems to code only for "circle on rough canvas". While this is definitely the defining characteristic of the contributing images, it is not helpful for stylization. The other rows are examples of degenerate archetypes, *i.e.* archetypes with a single contribution. (b) A non-sparse image decomposition, hence difficult to interpret. The strongest three components seem to represent the absence of texture, but it is not clear what their contribution is to the image style.

A.3 Additional Examples of Style Manipulation

In this section, we present additional examples of style enhancement and interplation, as well as examples of stylization of natural photographs.

(a) “Woman with Book” by Pablo Picasso. From the GanGogh collection.

Figure A.9 – We demonstrate the enhancement of the two most prominent archetypal styles for different artworks. The middle panel shows a near-perfect reconstruction of the original content image in every case and uses parameters $\gamma, \delta = 0.5$. Then, we increase the relative weight of the strongest component towards the left, and of the second component towards the right. Simultaneously, we increase γ and δ from 0.5 in the middle panel to 0.95 on the outside.

Figure A.10 – “Maria and Baby” by Robert Henri. Free archetypal combination.

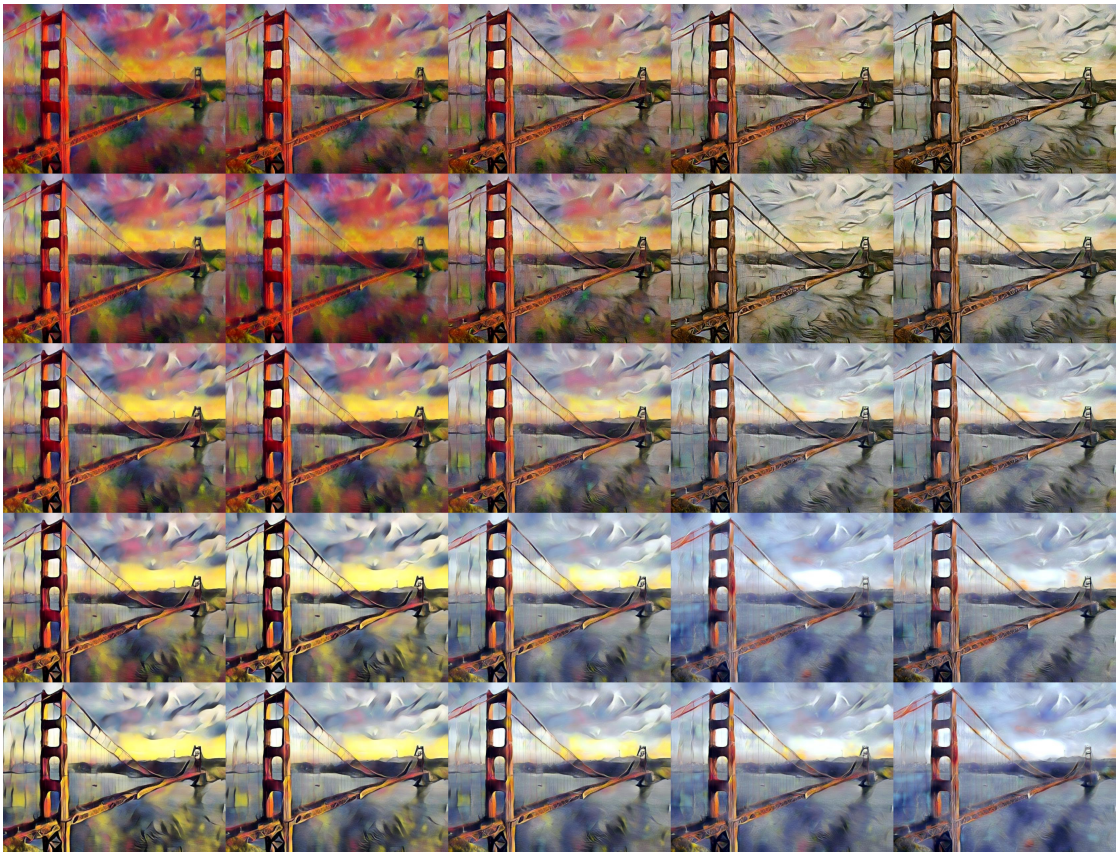


Figure A.11 – Golden Gate Bridge

Figure A.12 – Additional examples of style enhancements of van Gogh's works.

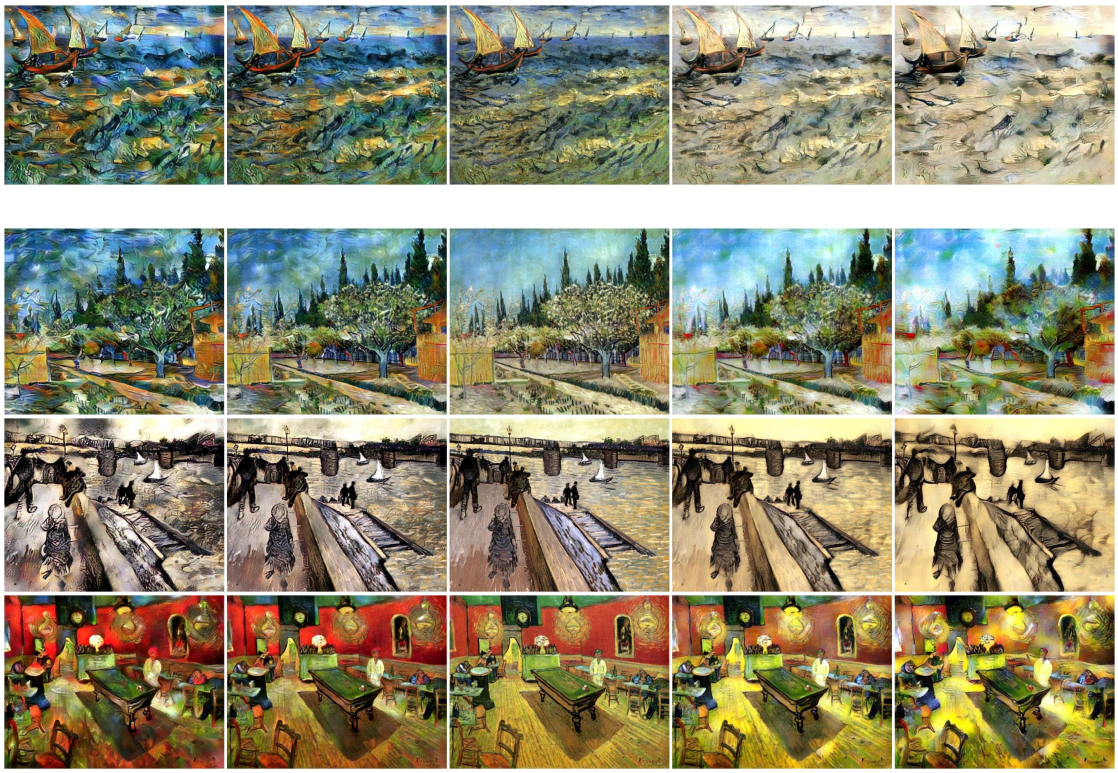
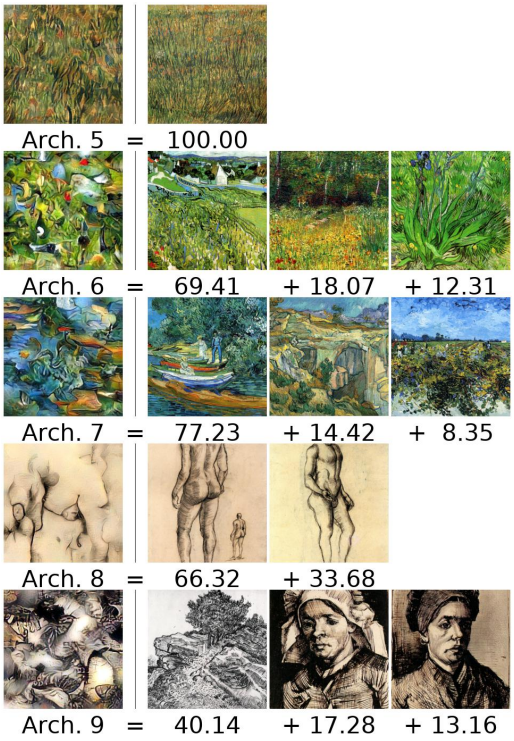


Figure A.13 – Additional examples of style enhancements of van Gogh's works.

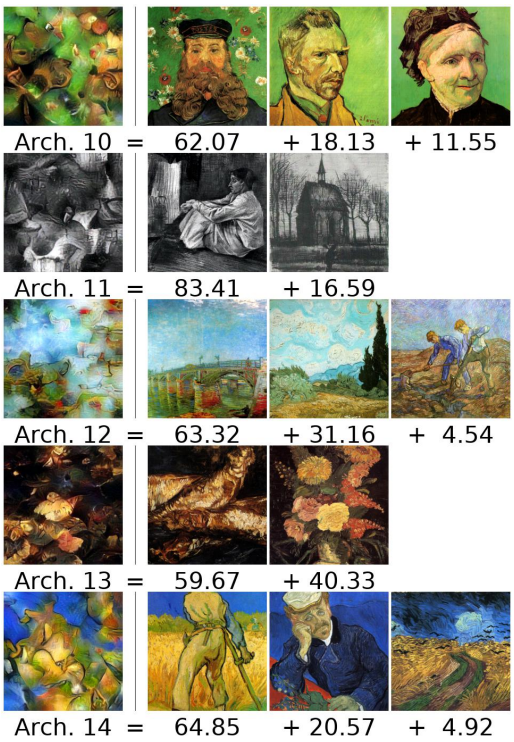
A.4 Full Set of van Gogh's Archetypes

In this section, we present the $k = 32$ archetypes learned on the collection of Van Gogh's paintings; the archetypes seem to cover van Gogh's artistic development relatively accurately. The full set of archetypes is shown in Figures A.14 and A.15.

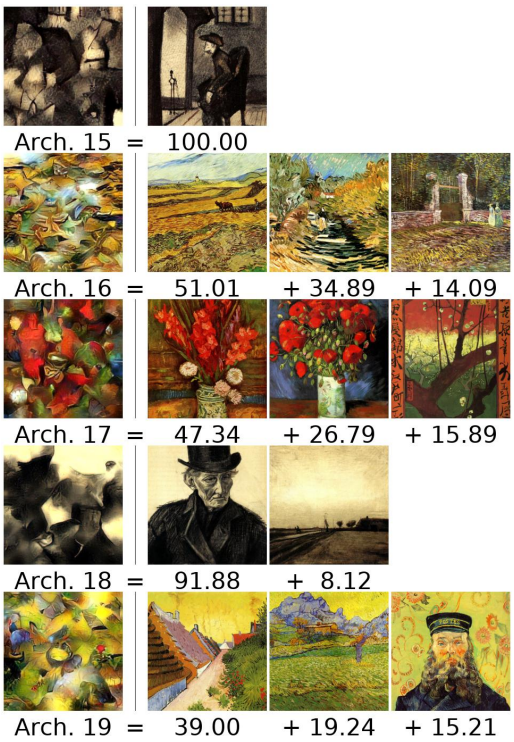
(a) Archetypes 0 to 4



(b) Archetypes 5 to 9

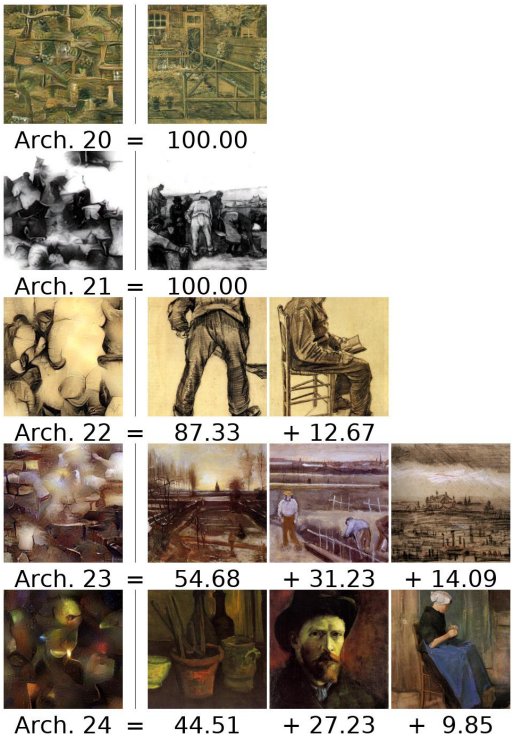


(c) Archetypes 10 to 14

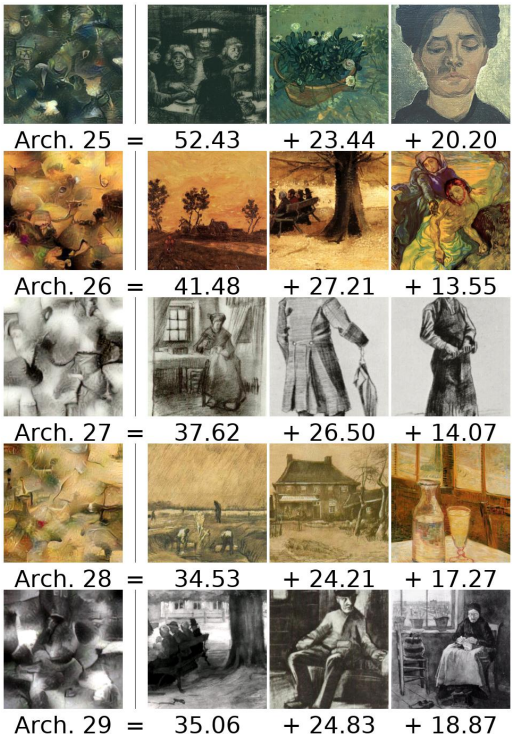


(d) Archetypes 15 to 19

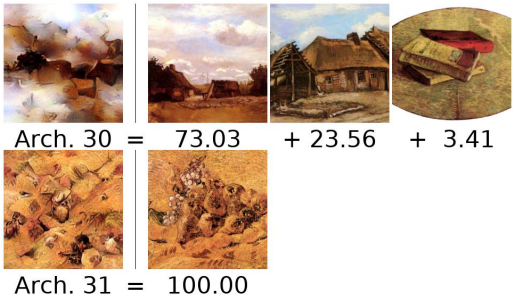
Figure A.14 – Archetypes 0 to 19



(a) Archetypes 20 to 24



(b) Archetypes 25 to 29



(c) Archetypes 30 to 31

Figure A.15 – Archetypes 20 to 31

Bibliography

[1] H.A. Aly and E. Dubois. "Image Up-Sampling Using Total-Variation Regularization with a New Observation Model". In: *IEEE Transactions on Image Processing* 14.10 (Oct. 2005), pp. 1647–1659. ISSN: 1941-0042. DOI: 10.1109/TIP.2005.851684.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. Jan. 26, 2017. URL: http://arxiv.org/abs/1701.07875 (visited on 01/30/2017).

[3] Christian Bauckhage and Kasra Manshaei. "Kernel Archetypal Analysis for Clustering Web Search Frequency Time Series". In: *2014 22nd International Conference on Pattern Recognition*. 2014 22nd International Conference on Pattern Recognition. Aug. 2014, pp. 1544–1549. DOI: 10.1109/ICPR.2014.274.

[4] Richard H. Byrd et al. "A Limited Memory Algorithm for Bound Constrained Optimization". In: *SIAM Journal on Scientific Computing* 16.5 (Sept. 1, 1995), pp. 1190–1208. ISSN: 1064-8275. DOI: 10.1137/0916069. URL: https://epubs.siam.org/doi/10.1137/0916069 (visited on 09/20/2020).

[5] Dongdong Chen et al. "StyleBank: An Explicit Representation for Neural Image Style Transfer". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[6] Tian Qi Chen and Mark Schmidt. *Fast Patch-Based Style Transfer of Arbitrary Style*. Dec. 13, 2016. URL: http://arxiv.org/abs/1612.04337 (visited on 04/18/2018).

[7] Yuansi Chen, Julien Mairal, and Zaid Harchaoui. "Fast and Robust Archetypal Analysis for Representation Learning". In: CVPR 2014 - IEEE Conference on Computer Vision & Pattern Recognition. IEEE, June 24, 2014, pp. 1478–

1485. DOI: 10.1109/CVPR.2014.192. URL: https://hal.inria.fr/hal-00995911/document (visited on 05/10/2017).

[8] Casey Chu, Andrey Zhmoginov, and Mark Sandler. *CycleGAN, a Master of Steganography*. Dec. 8, 2017. URL: http://arxiv.org/abs/1712.02950 (visited on 02/27/2018).

[9] Eric Chu. *Artistic Influence GAN*. Dec. 8, 2018. URL: https://nips2018creativity.github.io/doc/Artist_Influencers.pdf (visited on 12/09/2018).

[10] Adele Cutler and Leo Breiman. "Archetypal Analysis". In: *Technometrics* 36.4 (1994), pp. 338–347. ISSN: 0040-1706. DOI: 10.2307/1269949.

[11] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[12] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. *A Learned Representation For Artistic Style*. Oct. 24, 2016. URL: http://arxiv.org/abs/1610.07629 (visited on 05/02/2018).

[13] *Edmond de Belamy - Obvious Art*. URL: http://obvious-art.com/edmond-de-belamy.html (visited on 09/01/2019).

[14] A. A. Efros and T. K. Leung. "Texture Synthesis by Non-Parametric Sampling". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2. 1999, 1033–1038 vol.2. DOI: 10.1109/ICCV.1999.790383.

[15] Alexei A. Efros and William T. Freeman. "Image Quilting for Texture Synthesis and Transfer". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 341–346. ISBN: 978-1-58113-374-5. DOI: 10.1145/383259.383296. URL: http://doi.acm.org/10.1145/383259.383296 (visited on 08/30/2019).

[16] Ahmed Elgammal et al. *CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms*. June 21, 2017. URL: http://arxiv.org/abs/1706.07068 (visited on 05/10/2018).

[17] Jakub Fišer et al. "StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings". In: *ACM Trans. Graph.* 35.4 (July 2016), 92:1–92:11. ISSN: 0730-0301. DOI: 10.1145/2897824.2925948. URL: http://doi.acm.org/10.1145/2897824.2925948 (visited on 07/11/2019).

[18] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A Neural Algorithm of Artistic Style*. Aug. 26, 2015. URL: http://arxiv.org/abs/1508.06576.

[19] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *Texture Synthesis Using Convolutional Neural Networks*. May 27, 2015. URL: http://arxiv.org/abs/1505.07376 (visited on 06/14/2019).

[20] Leon A. Gatys et al. *Controlling Perceptual Factors in Neural Style Transfer*. Nov. 23, 2016. URL: http://arxiv.org/abs/1611.07865 (visited on 11/10/2017).

[21] Golnaz Ghiasi et al. *Exploring the Structure of a Real-Time, Arbitrary Neural Artistic Stylization Network*. May 18, 2017. URL: http://arxiv.org/abs/1705.06830 (visited on 06/14/2017).

[22] Andrew Gilbert et al. "Disentangling Structure and Aesthetics for Style-Aware Image Completion". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 1848–1856. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Gilbert_Disentangling_Structure_and_CVPR_2018_paper.html (visited on 09/10/2018).

[23] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf (visited on 09/01/2019).

[24] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. Mar. 31, 2017. URL: http://arxiv.org/abs/1704.00028 (visited on 04/10/2017).

[25] David J. Heeger and James R. Bergen. "Pyramid-Based Texture Analysis/Synthesis". In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 229–238. ISBN: 978-0-89791-701-8. DOI: 10.1145/218380.218446. URL: http://doi.acm.org/10.1145/218380.218446 (visited on 05/11/2018).

[26] Aaron Hertzmann. "Painterly Rendering with Curved Brush Strokes of Multiple Sizes". In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 453–460. ISBN: 978-0-89791-999-9. DOI: 10.1145/280814.280951. URL: http://doi.acm.org/10.1145/280814.280951 (visited on 05/11/2018).

[27] Aaron Hertzmann et al. "Image Analogies". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 327–340. ISBN: 978-1-58113-374-5. DOI: 10.1145/383259.383295. URL: http://doi.acm.org/10.1145/383259.383295 (visited on 05/31/2018).

[28] Elad Hoffer and Nir Ailon. *Deep Metric Learning Using Triplet Network*. Version 4. 2015. URL: http://arxiv.org/abs/1412.6622 (visited on 01/28/2020).

[29] D. L. Hoffmann et al. "U-Th Dating of Carbonate Crusts Reveals Neandertal Origin of Iberian Cave Art". In: *Science* 359.6378 (Feb. 23, 2018), pp. 912–915. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aap7778. URL: https://science.sciencemag.org/content/359/6378/912 (visited on 09/20/2020).

[30] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization*. Mar. 20, 2017. URL: http://arxiv.org/abs/1703.06868 (visited on 11/06/2017).

[31] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Feb. 10, 2015. URL: http://arxiv.org/abs/1502.03167 (visited on 01/09/2017).

[32] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. Nov. 21, 2016. URL: http://arxiv.org/abs/1611.07004 (visited on 02/22/2017).

[33] Ondřej Jamriška et al. "Stylizing Video by Example". In: *ACM Trans. Graph.* 38.4 (July 2019), 107:1–107:11. ISSN: 0730-0301. DOI: 10.1145/3306346.3323006. URL: http://doi.acm.org/10.1145/3306346.3323006 (visited on 08/19/2019).

[34] Nikolay Jetchev, Urs Bergmann, and Gokhan Yildirim. *Copy the Old or Paint Anew? An Adversarial Framework for (Non-) Parametric Image Stylization*. Nov. 22, 2018. URL: http://arxiv.org/abs/1811.09236 (visited on 12/08/2018).

[35] John W. Selfridge. *Pablo Picasso*. Chelsea House Pub (Library), 1993. ISBN: 0-7910-1996-9. URL: https://www.amazon.com/Pablo-Picasso/dp/0791019969?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0791019969.

[36] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*. Mar. 26, 2016. URL: http://arxiv.org/abs/1603.08155 (visited on 03/07/2018).

[37] Kenny Jones and Derrick Bonafilia. *GANGogh: Creating Art with GANs*. URL: https://towardsdatascience.com/gangogh-creating-art-with-gans-8d087d8f74a1 (visited on 03/15/2018).

[38] B. Julesz. "Visual Pattern Discrimination". In: *IRE Transactions on Information Theory* 8.2 (Feb. 1962), pp. 84–92. ISSN: 0096-1000. DOI: 10.1109/TIT.1962.1057698.

[39] Sergey Karayev et al. *Recognizing Image Style*. Nov. 14, 2013. URL: http://arxiv.org/abs/1311.3715 (visited on 03/02/2018).

[40] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. Dec. 12, 2018. URL: http://arxiv.org/abs/1812.04948 (visited on 02/13/2019).

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (visited on 07/21/2019).

[42] Carl Lagoze and Herbert Van de Sompel. "The Open Archives Initiative: Building a Low-Barrier Interoperability Framework". In: *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries* (Roanoke, Virginia, USA). JCDL '01. New York, NY, USA: ACM, 2001, pp. 54–62. ISBN: 978-1-58113-345-5. DOI: 10.1145/379437.379449. URL: http://doi.acm.org/10.1145/379437.379449 (visited on 06/07/2019).

[43] Yanghao Li et al. "Demystifying Neural Style Transfer". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI'17. Melbourne, Australia: AAAI Press, 2017, pp. 2230–2236. ISBN: 978-0-9992411-0-3. URL: http://dl.acm.org/citation.cfm?id=3172077.3172198.

[44] Yijun Li et al. *Universal Style Transfer via Feature Transforms*. May 23, 2017. URL: http://arxiv.org/abs/1705.08086 (visited on 02/22/2018).

[45] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. Feb. 20, 2015. URL: http://arxiv.org/abs/1405.0312 (visited on 01/27/2020).

[46] Thomas Edward Lombardi. "The Classification of Style in Fine-Art Painting". PhD Thesis. New York, NY, USA: Pace University, 2005.

[47] D.G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2. Sept. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[48] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data Using T-SNE". In: *Journal of Machine Learning Research* 9 (Nov 2008), pp. 2579–2605. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html (visited on 05/10/2018).

[49] Julien Mairal, Francis Bach, and Jean Ponce. *Sparse Modeling for Image and Vision Processing*. Nov. 12, 2014. URL: http://arxiv.org/abs/1411.3230 (visited on 06/17/2019).

[50] Julien Mairal et al. *Online Learning for Matrix Factorization and Sparse Coding*. Aug. 1, 2009. URL: http://arxiv.org/abs/0908.0050 (visited on 06/17/2019).

[51] Shin Matsuo and Keiji Yanai. "CNN-Based Style Vector for Style Image Retrieval". In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ICMR '16. New York, NY, USA: ACM, 2016, pp. 309–312. ISBN: 978-1-4503-4359-6. DOI: 10.1145/2911996.2912057. URL: http://doi.acm.org/10.1145/2911996.2912057 (visited on 12/09/2018).

[52] Takeru Miyato et al. *Spectral Normalization for Generative Adversarial Networks*. Feb. 16, 2018. URL: http://arxiv.org/abs/1802.05957 (visited on 02/26/2018).

[53] Alexander Mordvintsev et al. "Differentiable Image Parameterizations". In: *Distill* 3.7 (July 25, 2018), e12. ISSN: 2476-0757. DOI: 10.23915/distill.00012. URL: https://distill.pub/2018/differentiable-parameterizations (visited on 10/04/2020).

[54] Katherine Nash and Richard H. Williams. "Computer Program for Artists: ART 1". In: *Leonardo* 3.4 (1970), pp. 439–442. ISSN: 0024-094X. DOI: 10.2307/1572264.

[55] Adam Paszke et al. "Automatic Differentiation in PyTorch". In: (Oct. 28, 2017). URL: https://openreview.net/forum?id=BJJsrmfCZ (visited on 06/14/2019).

[56] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct. 2011), pp. 2825–2830. ISSN: 1533-7928. URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html (visited on 06/14/2019).

[57] Paraskevas Pegios, Nikolaos Passalis, and Anastasios Tefas. *Style Decomposition for Improved Neural Style Transfer*. Nov. 30, 2018. URL: http://arxiv.org/abs/1811.12704 (visited on 02/25/2019).

[58] Ryan Poplin and Adam Prins. *Behind the Scenes with Stadia's Style Transfer ML*. May 18, 2019. URL: https://stadia.dev/intl/en/blog/behind-the-scenes-with-stadias-style-transfer-ml/ (visited on 01/28/2020).

[59] Javier Portilla and Eero P. Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients". In: *International Journal of Computer Vision* 40.1 (Oct. 1, 2000), pp. 49–70. ISSN: 1573-1405. DOI: 10.1023/A:1026553619983. URL: https://doi.org/10.1023/A:1026553619983 (visited on 02/05/2019).

[60] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. Nov. 19, 2015. URL: http://arxiv.org/abs/1511.06434 (visited on 09/01/2019).

[61] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Dec. 1, 2015), pp. 211–252. ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: https://doi.org/10.1007/s11263-015-0816-y (visited on 07/21/2019).

[62] Artsiom Sanakoyeu et al. *A Style-Aware Content Loss for Real-Time HD Style Transfer*. July 26, 2018. URL: http://arxiv.org/abs/1807.10201 (visited on 03/06/2019).

[63] Lior Shamir and Jane A. Tarakhovsky. "Computer Analysis of Art". In: *J. Comput. Cult. Herit.* 5.2 (Aug. 2012), 7:1–7:11. ISSN: 1556-4673. DOI: 10.1145/2307723.2307726. URL: http://doi.acm.org/10.1145/2307723.2307726 (visited on 06/21/2018).

[64] Lu Sheng et al. *Avatar-Net: Multi-Scale Zero-Shot Style Transfer by Feature Decoration*. May 10, 2018. URL: http://arxiv.org/abs/1805.03857 (visited on 02/13/2019).

[65] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Sept. 4, 2014. URL: http://arxiv.org/abs/1409.1556 (visited on 03/12/2018).

[66] Daniel Sýkora et al. *StyleBlit: Fast Example-Based Stylization with Local Guidance*. July 9, 2018. URL: http://arxiv.org/abs/1807.03249 (visited on 12/08/2018).

[67] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[68] Ondřej Texler et al. *Enhancing Neural Style Transfer Using Patch-Based Synthesis*. The Eurographics Association, 2019. ISBN: 978-3-03868-078-9. DOI: 10.2312/exp.20191075. URL: https://diglib.eg.org:443/xmlui/handle/10.2312/exp20191075 (visited on 08/19/2019).

[69] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis*. Jan. 9, 2017. URL: http://arxiv.org/abs/1701.02096 (visited on 05/11/2018).

[70] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization*. July 27, 2016. URL: http://arxiv.org/abs/1607.08022 (visited on 05/11/2018).

[71] Dmitry Ulyanov et al. *Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images*. Mar. 10, 2016. URL: http://arxiv.org/abs/1603.03417 (visited on 04/17/2018).

[72] Paul Upchurch et al. *Deep Feature Interpolation for Image Content Changes*. Nov. 16, 2016. URL: http://arxiv.org/abs/1611.05507.

[73] Giorgio Vasari. *The Lives of the Artists (Oxford World's Classics)*. Trans. by Julia Conaway Bondanella and Peter Bondanella. Oxford University Press, 1998. ISBN: 0-19-283410-X.

[74] Michael J. Wilber et al. *BAM! The Behance Artistic Media Dataset for Recognition Beyond Photography*. Apr. 27, 2017. URL: http://arxiv.org/abs/1704.08614 (visited on 09/10/2018).

[75] Daan Wynen, Cordelia Schmid, and Julien Mairal. *Unsupervised Learning of Artistic Styles with Archetypal Style Analysis*. May 28, 2018. URL: http://arxiv.org/abs/1805.11155 (visited on 05/30/2018).

[76] Jianchao Yang et al. "Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition. June 2009, pp. 1794–1801. DOI: 10.1109/CVPR.2009.5206757.

[77] Mao-Chuang Yeh and Shuai Tang. *Improved Style Transfer by Respecting Inter-Layer Correlations*. Jan. 5, 2018. URL: http://arxiv.org/abs/1801.01933 (visited on 03/07/2018).

[78] N. Y.) Metropolitan Museum of Art (New York. *Italian Paintings: Venetian School;: A Catalogue of the Collection of the Metropolitan Museum of Art*. Distributed by New York Graphic Society, Greenwich, Conn, 1973. ISBN: 0-87099-079-9. URL: https://www.amazon.com/Italian-paintings-catalogue-collection-Metropolitan/dp/0870990799?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0870990799.

[79] Hang Zhang and Kristin Dana. *Multi-Style Generative Network for Real-Time Transfer*. Mar. 20, 2017. URL: http://arxiv.org/abs/1703.06953 (visited on 02/13/2019).

[80] Yang Zhou et al. *Non-Stationary Texture Synthesis by Adversarial Expansion*. May 11, 2018. URL: http://arxiv.org/abs/1805.04487 (visited on 05/17/2018).

[81] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks*. Mar. 30, 2017. URL: http://arxiv.org/abs/1703.10593 (visited on 01/08/2018).