UC Santa Cruz
UC Santa Cruz Electronic Theses and Dissertations

Title

Interactive Style Transfer for Data Visualization and Data Art

Permalink

https://escholarship.org/uc/item/66s3m0pk

Author

Dubey, Mahika

Publication Date

2019

Copyright Information

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

SANTA CRUZ

INTERACTIVE STYLE TRANSFER FOR DATA VISUALIZATION AND DATA ART

A thesis submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

COMPUTATIONAL MEDIA

by

Mahika Dubey

December 2019

The Thesis of Mahika Dubey
is approved:

_____

Angus G. Forbes, Chair

_____

David Lee

_____

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Interactive Style Transfer for Data Visualization and Data Art

by

Mahika Dubey

This thesis discusses *Data Brushes*, an interactive web application to explore neural style transfer using models trained on artistic data visualizations. The application invites casual creators to engage with deep convolutional neural networks to co-create custom artworks with a focus on style transfer networks created from canonical and contemporary works of data visualization and data art to demonstrate the versatility and flexibility of the algorithm. In addition to enabling a novel creative workflow, the process of interactively modifying an image via multiple style transfer networks reveals meaningful features encoded within the networks, and provides insight into the effects particular networks have on different images, or different regions within a single image. To evaluate *Data Brushes*, we gathered expert feedback from participants of a data science symposium and ran an observational study, finding that our application facilitates the creative exploration of neural style transfer for data art and enhances user intuition regarding the expressive range of style transfer features. This thesis explores both the practical uses of such tools for artists as *Data Brushes* and the interpretive uses of creating such venues for accessibility to computational art, remixing the purpose of data visualizations to be more than just graphical representations of information.

To my dad,

Abhay Dubey,

my personal and professional compass.

Acknowledgments

This thesis is a culmination of a research project developed over the course of my final few quarters at UC Santa Cruz. The successful completion of my degree would not have been possible without the support of my advisor, Professor Angus Forbes. I am grateful to have had the opportunity to work in the Creative Coding lab with him. Thank you Angus for helping me shape my interests and ideas into viable research projects and publications. I would also like to thank my lab mates Jasmine Otto, Sarah Frost, David Abramov, Ran Xu, Manu Mathew Thomas, Oskar Elek and Cassia Artanegara. Your company made graduate school feel like being on a team. I am also thankful to Professor David Lee for being on my committee and providing useful feedback on this thesis.

I would not have been able to finish my Masters degree without the love and encouragement of my family. My parents, Abhay and Smita, were my consistent champions through any challenges that came my way. I cannot express enough how much I value the level of independence and confidence they have instilled in me, while always being open to guiding me through tough moments. My sister Ambika has been my sounding board throughout my life, and though she was miles away in Seattle, the many video calls and visits up and down the coast ensured I always had a new adventure to look forward to. I am grateful to have had the chance to spend so much time with my brother Akash after years living away from home. While he never fails to remind me of my age, I can always rely on him to keep my spirits up. I would also like to thank my cousin Malveeka Tewari and brother-in-law Vivek Mishra for the advice through my

graduate school journey. It has truly been insightful to learn from the experience of two accomplished researchers throughout this process. And of course I cannot forget little Vedika, whose constant joy and abundant energy keeps me spontaneous.

Finally, there are a few friends I would like to thank outside of graduate school who really stand out over this past year. Bilal Gabula, for moving 15 minutes away and listening patiently through my many existential moments over the last few months. Years after our Urbana-Champaign days we find ourselves in the same neighborhood again! I owe much of my sanity to your friendship and consistent support. Ankur Mahesh, for the many dessert and chai runs all over the Bay Area. I hope to continue these traditions during your future graduate years. And Revathi Anil Kumar, for the ongoing conversations about life, learning, and the great outdoors. I can always count on your company when the wanderlust hits! Thank you for all the laughter and the reality checks in between.

The text of this thesis includes reprint of the following previously published paper, Data Brushes: Interactive Style Transfer for Data Art, Mahika Dubey, Jasmine Otto, and Angus G. Forbes. The work was published and presented at IEEE VIS at the Vis Arts Program in Vancouver, Canada, in October 2019. The The co-author listed in this publication directed and supervised the research which forms the basis for the thesis.

Part I

Background

Chapter 1

Introduction

Neural style transfer is an algorithm involving the encoding of an image-to-image function which minimizes some loss function relative to a source 'style image' for a given target 'content image'. In this way, a trained neural network is used to extract a feature space from a chosen source style image, and apply those features to the target content image, such that the content image is embellished with those features, thus emulating the style [17]. This thesis describes *Data Brushes*, a novel web-based application that encourages users to interactively explore the creative possibilities of style transfer in several ways. Neural style transfer was originally popularized through the imitation of iconic works by famous early 20th Century painters such as Van Gogh and Picasso, and *Data Brushes* further incorporate styles learned from seminal works of data visualization and analysis by Charles Minard and Edward Tufte, and from data art created by influential data designer Giorgia Lupi. The decision to focus on data art was made to promote visualization as a viable form of artistic expression, as well as to
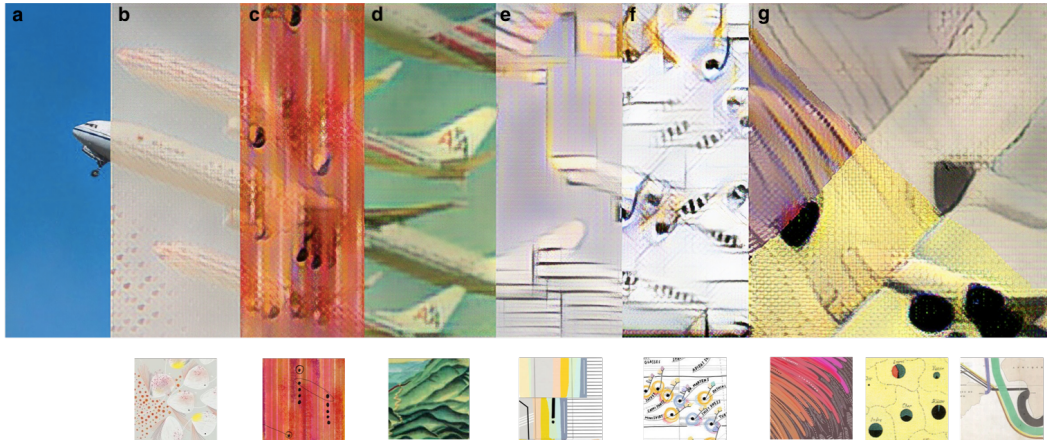
Figure 1.1: A series of style transfer data brushes applied to an image from Kelley's *Airportraits* project. **a** shows the original image without any styling, **b** shows the image styled with the *Bruises* brush (Lupi and King), **c** uses the *Hennessy* brush (Lupi, Maeda, and King), **d** is styled using our *Flatland* brush (generated from an image analyzed by Tufte), **e** is styled using a brush generated from one of Lupi's experimental scrapbook pieces, **f** showcases styling with the data brush based on Lupi's *Data Items*, and **g** demonstrates the result of layering multiple data brushes (including brushes created from visualizations by Minard).

generate discussion about additional extensions of style transfer applications. Figs. 1.1 and 1.2 show examples of a target image by Mike Kelley [13] manipulated using various styles transfer brushes generated by the application.

Data Brushes makes it easy to create complex data art collages by interacting with a digital canvas that transforms source images using predefined or custom data art styles. By facilitating creative manipulation, *Data Brushes* provides users— both casual creators [9] and computational researchers— with a way to interrogate the very feature sets isolated through the style transfer training process, enabling them to playfully interact with an AI medium, and creating a deliberate conflation of visual analysis and new media creation [37]. Two interactive modes, *magic markers* and *compositing stamps*, enable users to evaluate the effect of style transfer functions on their
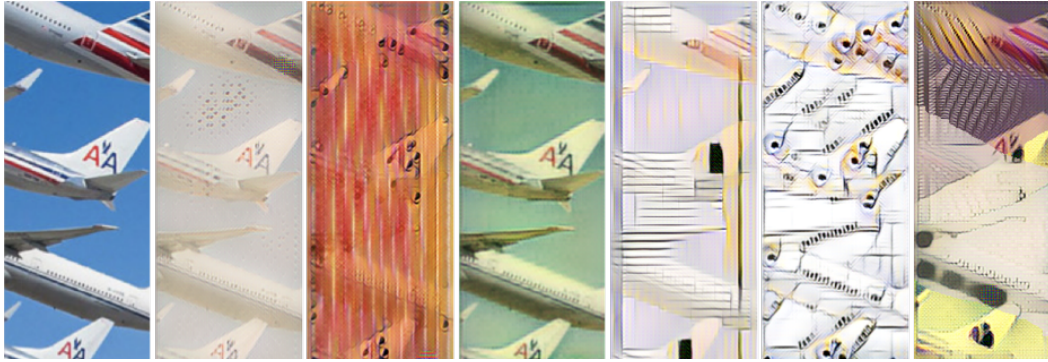
Figure 1.2: A comparison of the effects of styling the same target image with different data brushes. Here, we use the same brushes used in Fig. 1.1, but now each is applied to the exact same region of Kelley's photo (Fig.1.1d).

own images in different ranges of detail and localized effect, without requiring advanced technical or programming knowledge. The remaining chapters in Part I, Background, contextualize *Data Brushes* in terms of previous investigations of style transfer and sketch-based interfaces, including existing canvas-based applications for computational and machine learning art. Part II, Application Design, describes the developed system in detail, expanding on design decisions, the process of model training, and providing visual evidence to demonstrate the expressive range of our style brushes via the two available modes. And finally, Part III, Discussion and Analysis, summarizes the expert feedback collected from data scientists, presents details and insights from the observational study sessions, and discusses how *Data Brushes* facilitates new creative roles at the intersections of art, curation, and data visualization, introducing new questions that arose during the development of this project, and outlining plans for future iterations of the application.

Chapter 2

Related Work

2.1   Style Transfer

Style transfer refers to the method of integrating the distinct features of a source image into the existing content of a target image, transforming, to take a common example, a photograph of a landscape (the target image) into a new image that looks like an Impressionist painting (the source image), as depicted visually in Fig. 2.1. Inspired by the function and structure of biological neurons in the human brain, deep neural networks are constructed and trained to mimic visual recognition and understanding, and are trained to encode the visual features of a single source image [12, 23]. In general, a successful style transfer should transfer visual properties of the source image without altering the spatial layout and structural information of the target image it is being applied to. The alternating convolutional and subsampling layers of the models reproduce strokes, palette choices, and other perceptual properties of the style image,
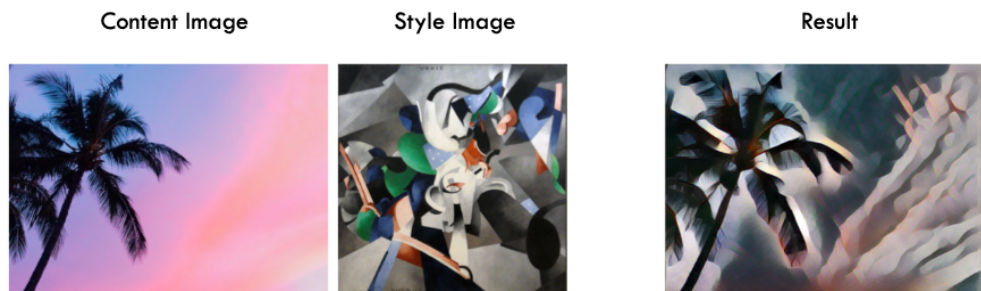
Figure 2.1: A simple example demonstrating the result of style transfer using a content image of a palm tree at sunset being rendered in the style of the target image, the painting *Udnie, Young American Girl* by artist Francis Picabia [30]

in particular where similar features are present in or implied by the content image.

While multiple algorithms have been developed over the past few years, *Data Brushes* uses the approach pioneered by Gatys et al. [17], which encodes the feature space of a source image using the VGG 19 neural network, which has a deep architecture with 16 convolutional layers and 5 pooling layers [35]. The VGG 19 neural network is initially trained on ImageNet to learn general feature encoding, then generates a model that captures the specific features of the style image. This model is then used to transform user selected input regions on our digital canvas. Fig. 2.2 presents a flowchart that provides an overview of the process specifically used in *Data Brushes* for applying trained networks to new content.

2.2   Canvas Applications

With digital technologies and interfaces becoming increasingly rich, many canvas-based tools for artists are available, inspired by the pioneering work of Sutherland's *Sketchpad* [38], as well as by more contemporary systems [34, 29]. The direct

6
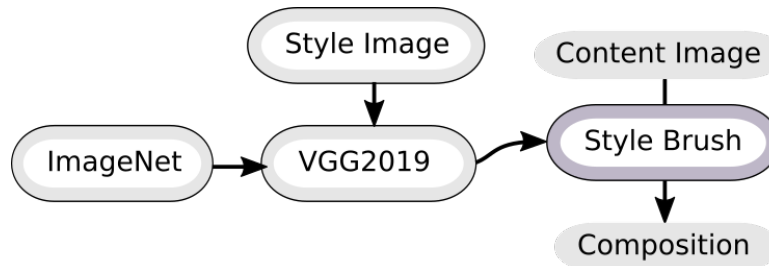
Figure 2.2: A high-level overview of the style transfer algorithm used in the *Data Brushes* application. Given a source 'style image', a model is trained through a VGG 19 network in order to generate an image-to-image function which will compress the primary features of any input 'content image' (minimizing loss relative to the style image), and then expand the content image into a new composition that retains the overall spatial layout of the target image, while adding the stylistic elements learned from the source image.

manipulation affordance promotes user interaction with abstract functions, extending traditional mediums of 2D art making. These kinds of interactions closely follow the concept of a 'magic lens' in which clicking actions are used to place small filters on digital images to change the underlying content [5]. Yet despite the simplicity of such interactions, computational art and creative coding are too often inaccessible to less technical users. This gap is reduced through the development of applications that make algorithms interactive, such as sketch-based interfaces that incorporate a pen or brush metaphor that lets users to manipulate pixels with ease [3, 15, 19, 22].

Jacobs et al.[20] explore this technique in their procedural illustration tool *Para*, providing a drawing environment for artists that includes interactive brushes with unique and customizable designs and styles, specifically adapting techniques (including symmetries and particle systems) from procedurally-generated art [16, 28]. Their expert evaluations support the usefulness of such a software system for artists to gain

control of algorithms. Expanding on tools for computer-assisted art, applications aimed at casual creators for photo editing and image filtering have been augmented using style transfer [33]. While many popular image effects such as color enhancement and vintage styling can be accomplished through simple pixel manipulations, style transfer can provide uncannily effective results when emulating an artist's work. For example, a recent mobile style transfer application by Reimann et al. [31] called *MaeSTrO* extends this style transfer functionality by enabling users to apply multiple styles onto distinct, user-selected, masked areas of an image. In *MaeSTrO*, the user is directed to produce a layer of semantic annotation, as is sometimes used in non-ML style transfer [11]. Bau et al.'s *GANPaint Studio* provides direct interaction with neural networks by using brushing on a canvas to insert or remove physical features from an image by manipulating underlying generative networks [4]. Their approach, unlike the one used in *Data Brushes*, is to apply texture synthesis over the entire canvas based, using spatially localized edits to the low-dimensional structure that are observed by a narrow, intermediate layer of their network. By contrast, neuronal activations (in the *compositing stamps* mode) are varied in *Data Brushes* by changing the local domain that the networks are exposed to.

These approaches, and others like them [21], have been used to develop applications that encourage the casual consumption of style transfer as a form of artistic expression. *Data Brushes* encourages users to create compositions interactively using a palette of style transfer networks, providing different modes for applying styles onto a single image. As described in later sections, the *Data Brushes* style transfer brushes are responsive both to the artist's manual input and the strokes and patterns observed

by a convolutional neural network.

Part II

Application Design

Chapter 3

Implementation and Interaction Modes

Data Brushes offers users two distinct modes that allow users to experiment with applying the styles of different famous data visualization artists onto an image of their own choosing. Fig. 3.1 presents an overview that illustrates the processing flow of each mode, which are described in more detail later in the thesis (in Chapters 3.1 and 3.2). The front-end interface, as shown in Fig. 3.2 makes use of layered HTML5 canvases as well as the D3.js library [6]. The back-end is built using ml5.js [1], a machine learning package built for the web on top of the TensorFlow.js library [36] that aims to improve the availability of common machine learning algorithms for developers and creative professionals of varying backgrounds. While ml5.js's pre-trained examples provide high quality results, the relationship of each painting to its 'perceptual loss', visible in the trained style model, can be difficult to interpret. In developing *Data Brushes*, this perceptual loss is exposed as a form of real-time evaluation of the effectiveness of a brush when applied to features in a particular target image, giving users a tool that is
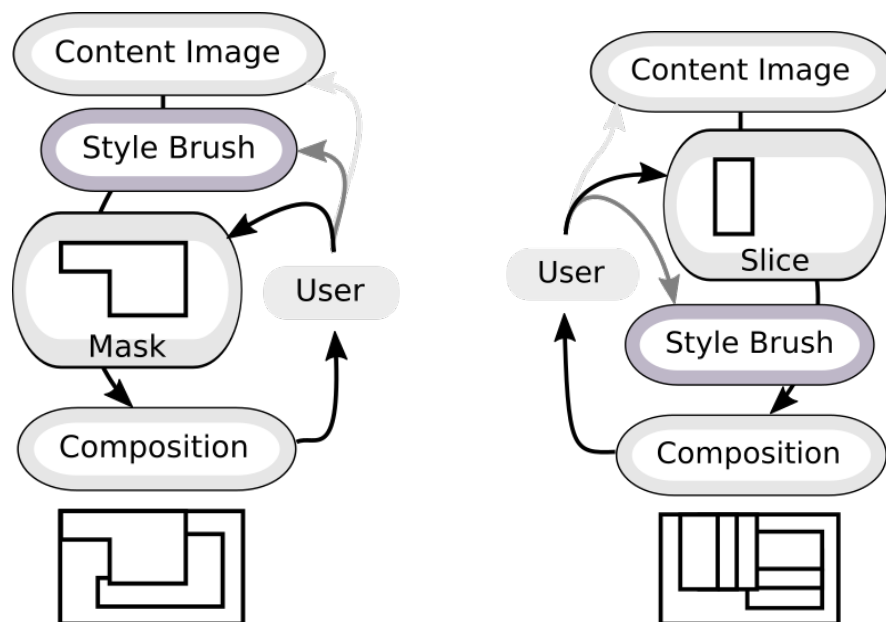
Figure 3.1: An overview of application design in each mode, as described in Chapters 3.1 and 3.2. The user is given control over style brushes of their choosing. Left: The user controls which regions of the composition it can effect, using a *magic marker* to reveal pre-computed styled layers underneath the image. Right: The user controls which regions it is affected via a *compositing stamp* that uses only features local to that selected area to transform the image in post, unaware of other features in the rest of the image. In both modes, the results are integrated into the content image, preserving their spatial context.

both exploratory and experiential, and making it possible to understand the mechanics

behind the feature extraction algorithm.

Thematically, *Data Brushes* pushes beyond traditional painterly style transfer

by instead training on data art made by data visualization experts and artists, including

Giorgia Lupi, Charles Minard, and Edward Tufte, as displayed in Fig. 3.3 (and described

in more detail in a following chapter on default brushes). Using the GPU cluster on

Paperspace's cloud infrastructure, the initial training of a network to encapsulate the

features of each image takes approximately 4-6 hours, depending on the size of the input.
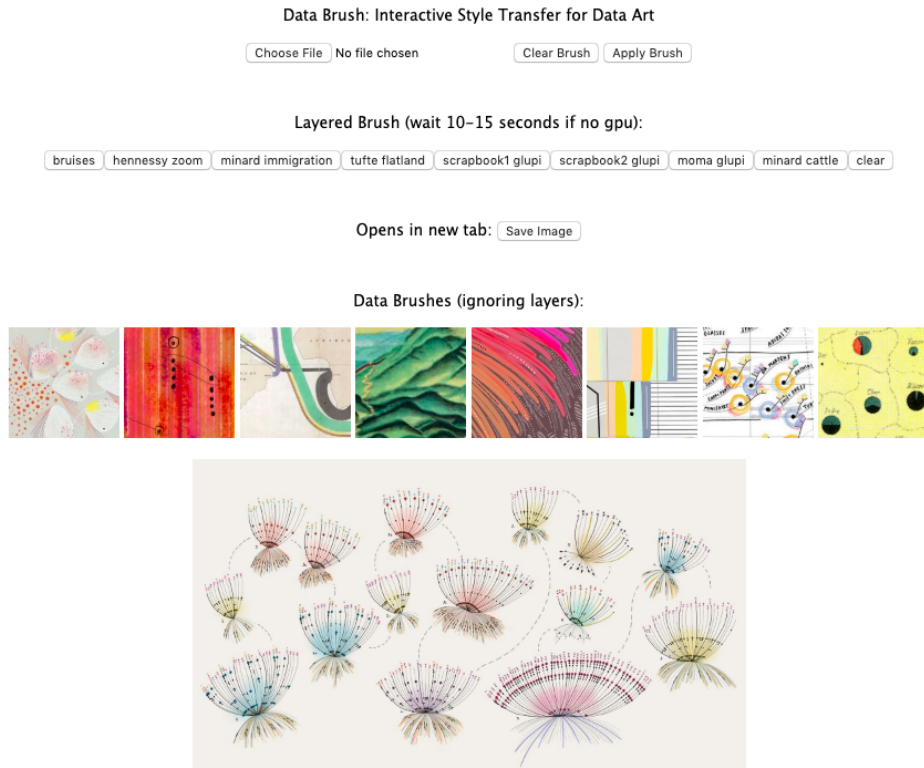
Figure 3.2: A screenshot of the interface of *Data Brushes* in *magic markers* mode.

The system lets users create new style transfer networks (i.e., new data brushes), and once trained, the networks can be queried within a few milliseconds, enabling interactive exploration. While each of the two modes provides a slightly different set of interactions with the style models, both encourage the user to explore the subtle differences in applied styles. By observing how each brush tends to transform particular patterns, lines, or colors within the target image in characteristic ways, a user can become familiar with the behaviors of the brushes and can begin to use them more skillfully and expressively.
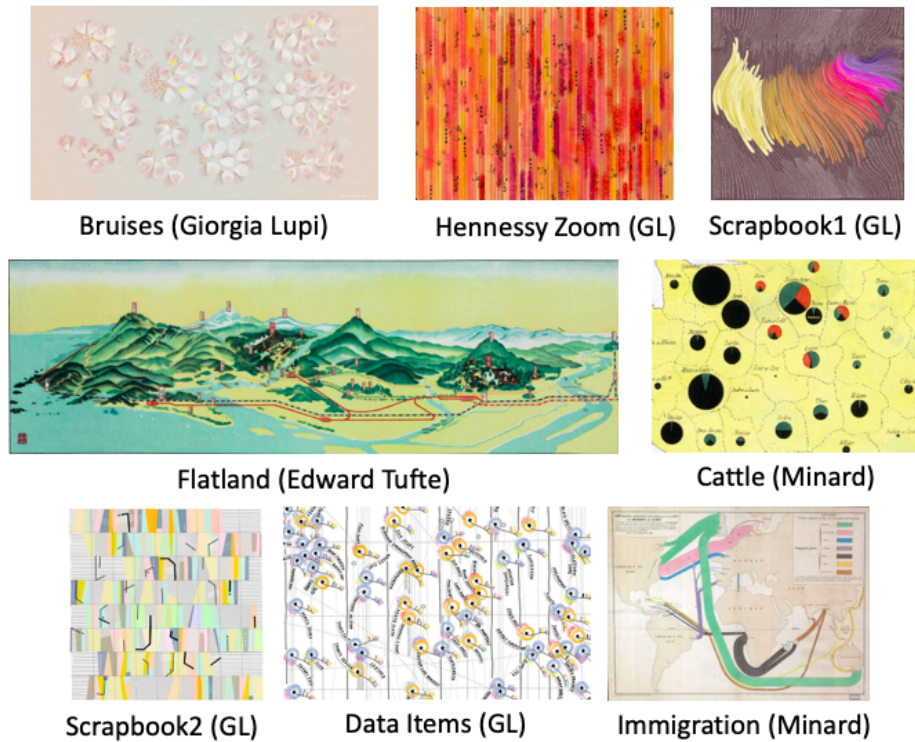
Figure 3.3: Examples of source images used to train some of our custom models. These original data visualizations and data art from Giorgia Lupi, Charles Minard, and Edward Tufte are used for a breadth of effects, and users can train their own custom style networks to add brushes to the *Data Brushes* application.

3.1 Magic Markers

The *magic markers* mode uses natural brush interactions so that users can 'paint' on different styles through simple selection and dragging. Usage of the application mimics the physical space of a painter with a palette of 'styles' (rather than colors), and includes undo functionality to revert recent changes to the canvas. A set of thumbnail images, as seen in Fig. 3.2, shows the available pre-trained models.

In this mode, all potential styled images are pre-computed and stored in hidden canvas layers aligned with the main canvas image. Clicking on one of the buttons copies

14
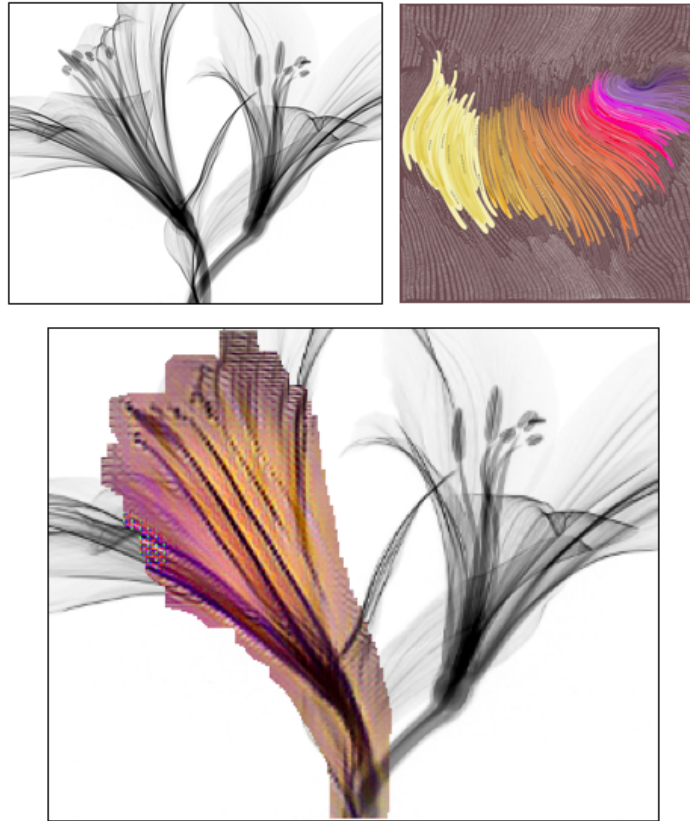
Figure 3.4: This figure illustrates the relationship between color and stoke features that can emerge when applying a style brush to an image. Using the *magic marker* mode with the source brush generated from the *Scrapbook1* image (top right), the coloration of the resulting image (bottom) is highly dependent on the direction of the lines in the black-and-white target image (top left). Orange shades begin to appear at the center of the flower, while the darker purples only appear as the angle of the flower's petals are close to the horizontal, replicating the exact usage of the colors as seen in the source image.

the styled image onto the hidden layer stored immediately below the main image. Brush selections can then be made on the canvas, and dragged around or reshaped as needed to reveal the styled image in the painted areas, unmasking the styled content by removing pixels from the top layer in real time. Painted sections are applied to the main canvas upon the click of the 'Apply Brush' button, which flatten the layered canvases into a single image. If changes have not been applied yet, the 'Clear Brush' can be used to remove any un-applied 'paint' by resetting the top canvas layer through the reversal of the pixel removal. Based on the varied size of the brush, users can change their level of details for different features, adding an additional level of customization to the co-creation process.

A limitation of solely implementing this approach is that multiple styles can not be layered on top of each other. Painting a new style over an already styled portion of the image replaces the content with the new style, instead of incorporating the effects of the old style onto the new styled image. Thus, the *magic markers* implementation also includes a multi-layering option to recursively style an image. Since this is inherently more computationally expensive, these multi-layering paint brushes can introduce a delay of up to 10 seconds when applying the style model to the hidden background canvas when using a consumer laptop without a dedicated GPU. However, this delay is greatly minimized on desktop computers with fast GPUs (less than 1 second on a workstation with an Nvidia GeForce RTX 2080 Ti). Once loaded, these layering brushes can style already-styled content, enabling a user to develop unique combinations of the features spaces of multiple different visualizations or artists. Figs. 1.1, 1.2, 3.4, 3.5, and
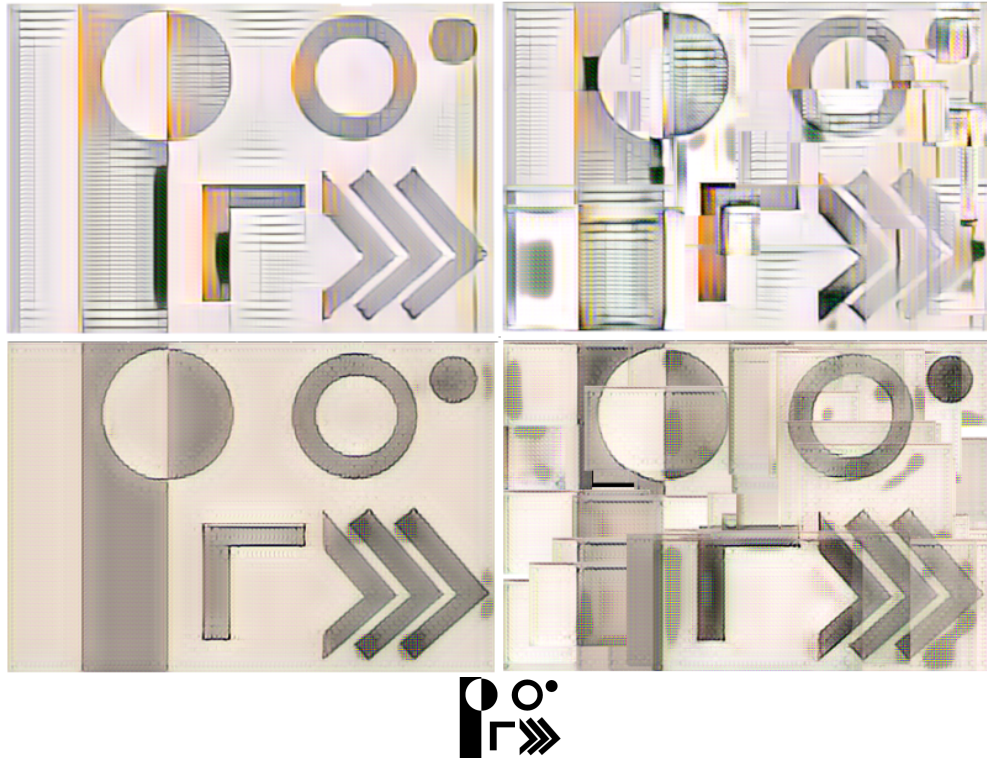
Figure 3.5: Comparison of *Scrapbook 2* (top row) and *Immigration* (bottom row) styles applied directly to the black-and-white test glyph at the bottom of the figure. The test glyph contains a range of geometric primitives in varying rotations, enabling us to observe how a style brush affects different features in a target image. The left column uses the *magic markers* mode with a single marker, applying each style to the entire image. The right column uses the *compositing stamps* mode with one stamp applied to the image many times, using the features local to the currently selected area. Note how the chevrons darken differently in each mode, how the 'ring' and 'moon' are transformed, how the *compositing stamps* mode introduces interesting border artifacts, and how overall there is an increased density of dramatic contrast which is introduced by our system as the style transfer network normalizes input values.

5.1 present examples created using *Data Brushes* in *magic markers* mode.

3.2   Compositing Stamps

In contrast to the pre-processing approach used in the *magic markers* mode, the *compositing stamps* mode uses instant post-processing of small selected areas of an image to layer 'style patches' to create interesting outputs resembling tiled mosaics or collages. Users can select different rectangular sections of their image of choice and immediately transform localized content based on the selected style. Unlike in the *magic markers* mode, image size is not scaled down in this mode, as the rapid selection of small rectangular areas reduces the computational load on the browser, especially on machines without GPU support.

Compositing stamps simulates the physical application of stamps colored with paint. Once a brush is selected, the user isolates a rectangular patch on the canvas through a click-and-drag operation. Upon mouse release, this patch will then be immediately replaced with a stylized version of the original content, applied only to that selected region. A patch of any size can be 'stamped'. A button labeled 'Apply Brush' merges the changes on the preview canvas to the main canvas, so that a user can then add additional layers of style on top of the transformed canvas. Pressing 'Clear Brush' removes the brush box from the canvas area, and pressing 'Clear Image' removes any changes on the preview layer that haven't yet been applied to the main canvas.

This mode is especially useful for interpreting what features are encapsulated

Figure 3.6: Here we show an example of how a user can probe the features of two of the default style brushes, *Scrapbook 2* (left) and *Immigration* (right), using the *compositing stamps* mode. A simple black-and-white test glyph is used as the target image (bottom), which contains simple symmetrical features. In the output shown on the left, horizontal 'grating' tends to occur on dark horizontal features, but will also sometimes occupy white areas where there is an absence of features. Interestingly, dark vertical lines are shaded orange in some regions, even though this color is not dominant in the original source image.

within a style transfer network, and for observing the interaction of style brushes on particular target images. By sending rectangular selections of the digital canvas directly to the style transfer algorithm, we can also study the trained network on a continuous deformation of content (see Fig. 3.6 for an example comparing two different brushes using this method of iterative deformation to accentuate features). This interactive process reveals the artifactual texture of the network, for instance in its response to boundaries and aspect ratios created by the successive application of small stamps. This mode exposes the style transfer network's expressive range when used to transform local regions. (See Fig. 3.5, right, which depicts an investigation of how smaller stamps generate interesting textures.)

3.3   Network Details

The models in this implementation of *Data Brushes* consist of a VGG 19 network comprising of 3 convolutional layers (representing kernels), 5 pairs of convolutional layers (representing a difference between kernels), 2 transpose layers (representing stamp-like patterns of application of kernels), 1 more convolutional layer, and a final activation and normalization step [35]. Every layer in the large 19 layer convolutional VGG network structure produces a different filter response. A representation of artistic style in an image is built by combining the correlations between these responses over the entire image [18]. Figs. 3.5 and 3.6 present examples of the trained networks applied to individual features of a basis image comprised of geometric primitives, demonstrating

the baseline behavior of the network. With no content image to warp, the produced imagery is a useful snapshot to understand the general color scheme and default stroke patterns used by the style.

Chapter 4

Default Brushes

Traditional style transfer interfaces often use famous works by painters with distinct, easily recognizable styles, such as Picasso, Kandinsky, and Van Gogh [17]. Given the motivation to improve accessibility to computational art for casual creators and artists, while also providing insight into the internal behavior of a style transfer network, *Data Brushes* provides users with brushes based on data visualization and data art works that emphasize the role of aesthetics and visual design in representing, communicating, and analyzing data. *Data Brushes* is an initial attempt at exploring these artistic features involved in visualizing data, demonstrating the flexibility of the style transfer algorithm and promoting the usage of such novel pieces in computational applications. Though the system supports the creation of style brushes from any image, Fig. 3.3 shows thumbnails of some of the images used to train the models provided by default in *Data Brushes*. The distinct visual features for these default brushes is explained below, providing context for the reasoning for choosing this selection:

- *Bruises, by Giorgia Lupi and Kaki King:* This piece tracks the progression and treatment of the daughter of Kaki King (a musician and long time collaborator of Giorgia Lupi) through an auto-immune disease called Idiopathic Thrombocytopenic Purpura (ITP) [26]. The piece is an example of empathetic data visualization, a key motivation in Giorgia Lupi's recent work in data humanism.

- *Hennessy, by Giorgia Lupi, John Maeda and Kaki King:* This visualization was part of a project to re-branding Hennessy's V.S.O.P Privilege on the 200th anniversary release of the cognac [27]. The piece combines information collected on Kaki's musical composition process during her trip to Cognac and traditional chalk patterns used in marking aging cognac barrels over time. Given the many interesting small details present in the image, we focused on a zoomed-in selection to train our data brush. The colorful vertical lines and the small notations and markings in black stood out as features most visible in styled content.

- *Scrapbook1, by Giorgia Lupi:* Giorgia Lupi spends significant time contemplating ideas for visual representations. These works are generally abstract, as she experiments with creative approaches to visualizing future datasets of interest, and are often posted by Lupi on her personal website and Instagram account [25]. This data brush produced the most varied colors, creating interesting and intuitive mapping of stroke direction to color [25].

- *Scrapbook2, by Giorgia Lupi:* Another creative data visualization sketched out by Giorgia Lupi in her set of Data Viz Scrapbooks [25]. We select the images we

felt produced the most interesting and new style features. This image used a lot of horizontal and vertical line elements, and the color associated with each are apparent when used on a content image with structural similarities.

- *Data Items, by Giorgia Lupi:* This piece is from a larger work made for an exhibition at the Museum of Modern Art in New York on the evolution of culturally important fashion and accessories over the last few centuries [24]. Similar to the Hennessy piece, we used a cropped selection of the original image so that our feature extractor could use the small details in the visualization. The colorful circles used to highlight items in the visualization were the primary item used to style any sort of content image, with the position of the text element being preserved, creating musical-note looking artifacts when brushing over new images.

- *Flatland, a piece examined by Edward Tufte:* This image was taken from Tufte's analysis of designing visualizations for flat interfaces such as paper or a screen [39]. He uses the image, a map from a Japanese travel guide, to demonstrate the unique usage of flat space in representing multi-dimensional information. The subtle blending used in this image produces an understated watercolor effect when applied to new content images, smoothing the image without changing its structure.

- *Cattle, by Charles Minard:* Much of Minard's work articulates patterns in the transportation of people and products. This particular image details the amount of cattle sent to Paris from various parts of France in 1858 [32]. We cropped out a part of the entire map visualization to capture specific features of interest. The

color difference between the base map and pie charts used in the visualization are captured by our feature extractor, leading to the drawing of dark circular areas over high frequency image content.

- *Immigration, by Charles Minard:* Another flow visualization by Minard, this piece highlights worldwide immigration in 1858, approximating people's origin and destination countries [32]. The antique map texture produced a sketchbook like effect when this brush was used to style images.

These pieces represent a breadth and evolution of data interpretation in the visualization community, and encode various levels of complexity in artifactual generation. The selected pieces are not meant to be a comprehensive representation of data visualization, however provide good variance in data content and visual style.

Part III

Discussion and Analysis

Chapter 5

Evaluation

The current version of *Data Brushes* was developed over a 10 month period, beginning in January 2019. During the development process, discussions held with artists, visualization researchers, and data scientists were used to collect feedback on the application and its use cases. This section summarizes two of these information gathering sessions, and the lessons learned from them.

5.1 Expert Feedback

A previous version of the *Data Brushes* application was presented at a Data Science Symposium held at University of California, Santa Cruz in early May 2019, with over 350 attendees, including faculty, graduate students, and data science experts from industry and research labs. An overview of the system architecture was presented alongside a range of example outputs. While participants were intrigued by the results, a repeatedly voiced concern was that the work was difficult to evaluate, as the application

is made to *support* the creative process, rather than to measure outputs against other style transfer techniques. The discussions with participants on the practicality and purpose of assessing creative support tools helped articulate that the project is a tool *for* evaluation rather than producing images *to be* evaluated, and that the outputs of *Data Brushes* function as a form of feature visualization themselves [7]. Another common thread in the feedback received was the interest in the use of data visualization pieces as inputs to the style networks, and it was validating to see an audience gain an appreciation for data visualization as a viable bridge between machine learning and creative arts.

5.2 User Study

Following the presentation at the Data Science Symposium, an audience of non-experts was selected for a study aimed at understanding how casual creators would interact with the application. An observational pilot study was conducted to investigate a user's ability to develop insight into the behavior of style transfer networks through the usage of *Data Brushes*.

The application focuses less on trying to generate an accurate or optimal style transfer algorithm, and more on the user experience itself. Since *Data Brushes* helps users make their own assessments of style brushes, it was interesting to recognize the insights that such users developed through the assisted interactive experience of the application. The pilot study was set up with 7 volunteers, in which each subject was
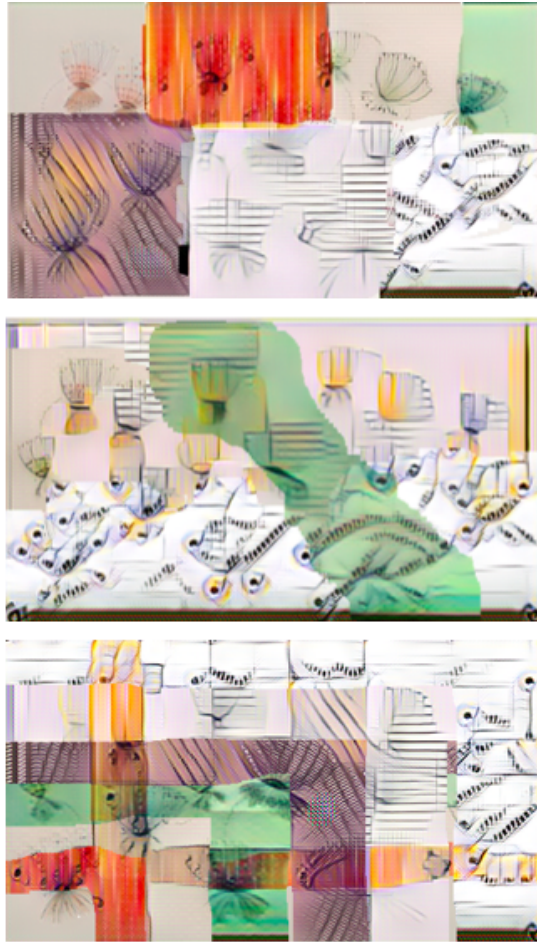
Figure 5.1: Examples of outputs created by users in our pilot study, using the *magic marker* mode for a single target image. One user (bottom) has generated a grid of all pairs of style brushes in order to characterize their interactions.

screen-recorded completing an unguided session of between 10 and 20 minutes of application exploration using the *magic markers* mode, followed by a short online survey that was filled out anonymously. Each user spent an additional 20 minutes completing the survey, and had access to the application during that time, in case they wished to revisit a particular aspect before finalizing an answer. Comments were collected indicating how users perceived the ease of use of the application, as well as remarks that
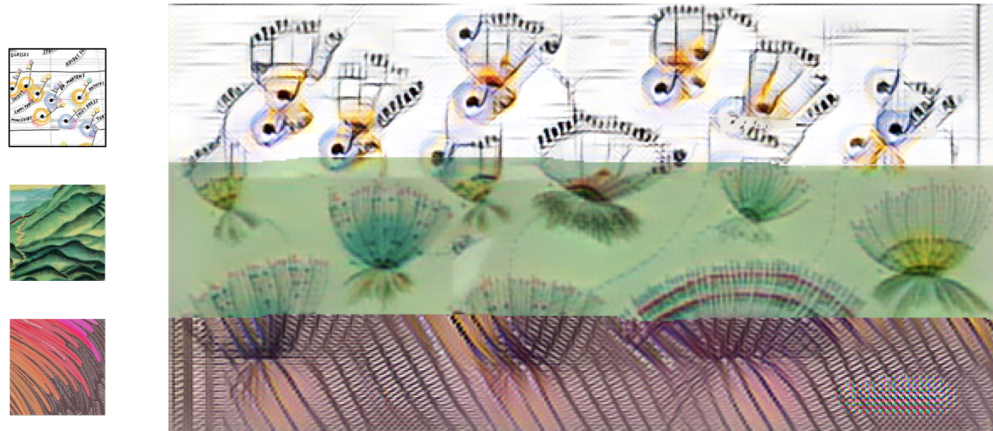
Figure 5.2: When asked to choose their favorite brushes, *Data Items*, *Flatland*, and *Scrapbook 1* were selected the most. The original style image is shown on the left of the main image, which has been styled by each brush respectively.

described how users interpreted the behavior of each data brush. The screen recordings of these interactions and the surveys were used to provide insight into the user experience, as well as to guide future plans for development of the application.

Fig. 5.1 presents 3 examples of the outputs created during the exploration sessions, in which participants were given a default target image and a selection of style transfer brushes. Given their freedom to layer brushes as they wished, it is interesting to note the different ways in which participants chose to combine all brushes for comparisons. Overall, the participants, all of who were new to neural style transfer, preferred the effects produced by brushes that generated a greater variance in patterns, as seen in Fig. 5.2, despite the fact that the use of these brushes were the most likely to obscure the underlying structural features of the target image. The primary reasoning provided for the preferences of these brushes was that they introduced more dramatic and unusual shapes and colors into the image, with one user explaining that "each of

Most frequent ranking for pattern generation

Least						Most

Average Rank	1.43	2.28	3.71	4.28	4.28

Most frequent ranking for content preservation

Worst						Best

Average Rank	1.43	2.00	2.57	4.14	4.85
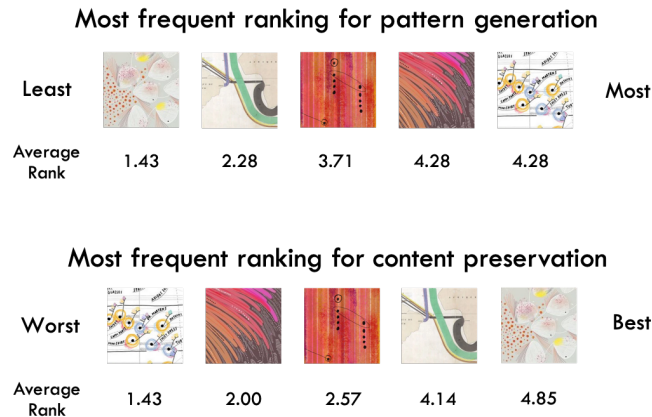
Figure 5.3: Users were asked to rank each data brush on a scale of 1 to 5 for pattern generation (1 representing the least variance in pattern generation and 5 representing the most variance) and content preservation (1 being the worst content preservation and 5 being the best). There is a visible correlation learned by our subjects between the variance in pattern generation and the ability of each brush to best preserve underlying content. We order the brush thumbnails in order, displaying the one which was most given that rank, and also show the average rank given to each of the brush.

these brushes has a distinctive style and character over and above the underlying image itself."

The survey was used to gauge the intuition developed through interaction with *Data Brushes*, and included questions that required users to rank brushes based on two criteria: the variance in pattern generation, and the level of content preservation. Fig. 5.3 summarizes the most frequent ranking provided by each user for both. Interestingly, there appears to be a correlation between user interpretation of the brushes and their ability either to generate varied patterns or to preserve the content image. This shows an initial indication that participants are able to glean insights into the behaviour of each brush, and provides motivation for developing further studies to investigate the potential benefits of demystifying computational methods for novice users.

Chapter 6

Data as Style

The development of this application generated an ongoing debate about the role of individual artistic vision in art creation. Does an artistic process begin with the intended content of the image and result in its 'style?' Does it begin with affordances of the medium (e.g., paints of various pigment, and brushes of various texture), and result in a 'form?' Or does creativity emerge precisely from the inherent tension between the two concerns? Style transfer architectures are designed to minimize the perceptual loss of an arbitrary image relative to a given style, transforming the content image into a compressed vector representation that encodes the salient features of the image [8]. By re-translating the vector representation and integrating it into another image, this generative pipeline allows a user to make creative decisions about which images to transform, making the algorithm itself an artistic medium. By embedding this procedure into an interactive canvas, users, even those with no knowledge of computer programming, also have to decide *where* to transform their images, and engage with

this new medium.

As programmers, we become *curators* of style brushes, and began to predict the pattern of responses that source images would yield, once fitted into the connection weights of a network pre-trained on visual primitives, for different target images. But the traditional roles of artist and curator are intertwined, because style transfer brushes depend likewise on a choice of style image. While general color schemes and structural motifs were reliably replicated (by the process of re-encoding a vector as an image), the style transfer brush's 'opinion' likewise depended on stroke directionality (as illustrated in Fig. 3.4), contrast, and aspect ratio (Fig. 3.6) in the content image. This instinct for the algorithm's valuation of various image segments is what users can hopefully take away from *Data Brushes*. Additionally, the aesthetic outcome of many data visualizations make them suitable choices for style source. And this type of creative experimentation could potentially help generate new approaches to the design of data visualization representations [14], including future style transfer applications that could, say, transform a simple scatter plot into a more evocative representation using interactive style brushes. We delve further into such ideas in the conclusion.

Chapter 7

Conclusion

Looking at future iterations of *Data Brushes*, there are some obvious technical as well as interpretive areas to expand on. From an application perspective, we consider extending the number of default brushes, and to make it easier for a user to train on style images of their choice, which could also help to facilitate data literacy [2], or at least provide an intuition for how features are encoded in neural style networks. A main challenge in accomplishing these tasks is optimizing for browser constraints and the existing bottleneck of training time, as we aim for a system that would allow immediate training of new data brushes based on user-submitted style images. As with any image manipulation task, style transfer is a computationally heavy process that scales with time as the size of the input increases. Despite these challenges, how can we better bridge the technical gap between such complex algorithms and the casual creators who could benefit from using them? And more generally. how could we improve accessibility for users of all experience levels for computational and algorithmic art?

Going beyond the practical usage of the application, we can contemplate the interdisciplinary uses that can evolve from such work. Data art is a result of aesthetic design choices and data transformations. In *Data Brushes*, we focus on the visual aspects of created pieces. However, could we develop a system that can train on the original data sets as well in order to learn some complex visualization schematics? Could specific brushes be created to cater to visualization needs of different domains, or perhaps could an entire version of this application be re-created for different fields? And conversely, could an understanding or intuition of style feature sets help inform unique data art pieces and infographics? Neural networks are used so often in highly creative ways, but if we can start to understand the way in which they encode features perhaps we can recognize important patterns and choices in visualization work.

The result of experimenting with web frameworks for machine learning in an artistic context, *Data Brushes* is an easy-to-use and explorable style transfer system for users of various levels of skill and creativity, promoting availability of creative tools for computational and algorithmic art. With this application, we move toward understanding the motivations of two very different audiences: researchers and content creators. While researchers seek to understand the mechanisms of content-aware image generation by deep convolutional networks, content creators desire a broader array of intuitive features to author and version their work. This application aims to satisfy the needs of both, so as to reduce the knowledge gap between the roles while promoting data visualization art, which intersects the two audiences. Providing users with the ability to rapidly generate custom brushes will increase access to artistic algorithms, improving the

understanding and accessibility of computational art for non-technical users. The *Data Brushes* web application is available at https://github.com/CreativeCodingLab/DataBrushes, along with source code, instructions, and video documentation.

Bibliography

[1] ml5js · friendly machine learning for the web.

[2] Basak Alper, Nathalie Henry Riche, Fanny Chevalier, Jeremy Boy, and Metin Sezgin. Visualization literacy at elementary school. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 5485–5497, 2017.

[3] Pascal Barla, Simon Breslav, Joëlle Thollot, François Sillion, and Lee Markosian. Stroke pattern analysis and synthesis. *Computer Graphics Forum*, 25(3):663–671, 2006.

[4] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics*, 38(4):59, 2019.

[5] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of the 20th Annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 73–80, 1993.

[6] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[7] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas.

[8] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv:1603.01768*, 2016.

[9] Katherine Compton. *Casual creators: Defining a genre of autotelic creativity support systems.* PhD thesis, UC Santa Cruz, 2019. https://escholarship.org/uc/item/4kg8g9gd.

[10] Mahika Dubey, Jasmine Otto, and Angus G Forbes. Data brushes: Interactive style transfer for data art. In *2019 IEEE VIS Arts Program (VISAP)*, pages 1–9. IEEE, 2019.

[11] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th ACM Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001.

[12] Michael Elad and Peyman Milanfar. Style transfer via texture synthesis. *IEEE Transactions on Image Processing*, 26(5):2338–2351, 2017.

[13] Nicholas Felton, Sven Ehmann, and Robert Klanten. *PhotoViz: Visualizing information through photography.* Gestalten, 2016.

[14] Angus G. Forbes. Articulating media arts activities in art-science contexts. *Leonardo*, 48:330–337, 2015.

[15] Angus G. Forbes, Tobias Höllerer, and George Legrady. Generative fluid profiles for interactive media arts projects. In *Proceedings of the ACM/Eurographics International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging (CAe)*, pages 37–43, 2013.

[16] Philip Galanter. What is generative art? Complexity theory as a context for art theory. In *Proceedings of the 6th Generative Art Conference*, 2003.

[17] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv:1508.06576*, 2015.

[18] Leon A. Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 262–270. 2015.

[19] Moritz Gerl and Tobias Isenberg. Interactive example-based hatching. *Computers & Graphics*, 37(1-2):65–80, 2013.

[20] Jennifer Jacobs, Joel Brandt, Radomír Mech, and Mitchel Resnick. Extending manual drawing practices with artist-centric programming tools. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 590:1–590:13, 2018.

[21] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *arXiv:1705.04058*, 2017.

[22] Joaquim Jorge and Faramarz Samavati. *Sketch-based interfaces and modeling.* Springer Science & Business Media, 2010.

[23] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv:1701.01036*, 2017.

[24] Giorgia Lupi. Data items: A fashion landscape at the museum of modern art. =http://giorgialupi.com/data-items-a-fashion-landscape-at-the-museum-of-modern-art, retrieved August 19, 2019.

[25] Giorgia Lupi. DataVizScrapbooks! http://giorgialupi.com, retrieved August 19, 2019.

[26] Giorgia Lupi and Kaki King. Bruises - the data we don't see. http://giorgialupi.com/bruises-the-data-we-dont-see, retrieved August 19, 2019.

[27] Giorgia Lupi, John Maeda, and Kaki King. Capturing the Blend. http://giorgialupi.com, retrieved August 19, 2019.

[28] Jon McCormack, Oliver Bown, Alan Dorin, Jonathan McCabe, Gordon Monro, and Mitchell Whitelaw. Ten questions concerning generative computer art. *Leonardo*, 47(2):135–141, 2014.

[29] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[30] Francis Picabia. Udnie (young american girl, the dance). Musée National d'Art Moderne, Centre Georges Pompidou, Paris, 1913. oil on canvas, 290x300 cm.

[31] M. Reimann, M. Klingbeil, S. Pasewaldt, A. Semmo, M. Trapp, and J. Döllner. MaeSTrO: A mobile app for style transfer orchestration using neural networks. In *Proceedings of the International Conference on Cyberworlds*, pages 9–16, Oct 2018.

[32] Sandra Rendgen. *The Minard system: The complete statistical graphics of Charles-Joseph Minard*. Chronicle Books, 2018.

[33] Amir Semmo, Tobias Isenberg, and Jürgen Döllner. Neural style transfer: A paradigm shift for image-based artistic rendering? In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, pages 5:1–13, 2017.

[34] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. Sketch based interfaces: Early processing for sketch understanding. In *Proceedings of the ACM Workshop on Perceptive User Interfaces*, pages 1–8, 2001.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[36] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, et al. TensorFlow.js: Machine learning for the Web and beyond. *arXiv:1901.05350*, 2019.

[37] Henry Staten. *Techne theory: A new language for art*. Bloomsbury Publishing, 2019.

[38] Ivan E Sutherland. Sketchpad: A man-machine graphical communication system. *Simulation*, 2(5):3–20, 1964.

[39] Edward R Tufte. *Envisioning information*. Graphics Press Cheshire, CT, 1993.