

AnimateSVG: Autonomous Creation and Aesthetics Evaluation of Scalable Vector Graphics Animations for the Case of Brand Logos

Deborah Mateja, Rebecca Armbruster, Jonathan Baumert, Tim Bleil, Jakob Langenbahn, Jan Christian Schwedhelm, Sarah Sester, Armin Heinzl

University of Mannheim
Mannheim, Germany

{mateja, armbruster, baumert, bleil, langenbahn, schwedhelm, sester, heinzl}@uni-mannheim.de

Abstract

In the light of the constant battle for attention on digital media, animating digital content plays an increasing role in modern graphic design. In this study, we use artificial intelligence methods to create aesthetic animations along the case of brand logos. With scalable vector graphics as the standard format in modern graphic design, we develop an autonomous end-to-end method using complex machine learning techniques to create brand logo animations as scalable vector graphics from scratch. We acquire data and setup a comprehensive animation space to create novel animations and evaluate them based on their aesthetics. We propose and compare two alternative computational models for automated logo animation and carefully weigh up their idiosyncrasies: on the one hand, we set up an aesthetics evaluation model to train an animation generator and, on the other hand, we combine tree ensembles with global optimization. Indeed, our proposed methods are capable of creating aesthetic logo animations, receiving an average rating of ‘good’ from observers.

Introduction

In modern user interface (UI) design, a constant battle for attention challenges designers. With UIs being ubiquitous, mobile applications and websites need to make their pages stand out. A recent trend is to bring graphics to life through motion designs, which raises attention and increases conversions in contrast to static content (Bruce, Murthi and Rao 2017). A logo serves as the “face” of a company and helps customers identify the brand (Walsh, Winterich, and Mittal 2010). Catching users’ attention through enjoyable logos in motion has the potential to make the brand more memorable.

In modern graphic design, the scalable vector graphic (SVG) format, as a vector-based image format for two-dimensional graphics, has established as a standard. This is mainly due to SVG’s scale invariance, i.e., they can be scaled to any size¹, with which they improve upon their rasterized counterparts, e.g., PNG and JPG format. SVGs document graphic designs in a tree-based structure consisting of

multiple elements. Each element represents a shape with its own style attributes like color or border. Research on using SVGs in Machine Learning (ML) remains scarce.

Working directly with SVG data is a core technological novelty of AnimateSVG, our proposed artificial intelligence (AI) application. Exemplified along the case of brand logos, we propose a model architecture for SVG animation, including data acquisition, construction of a comprehensive animation space, as well as generation and evaluation of the created animations. We implement and compare two computational approaches to SVG animation: First, we integrate a neural network evaluation model that approximates human ratings of logo animation aesthetics as a loss function into a neural network for the generation of novel SVG animations. Second, we combine tree-based models with global optimization to aesthetically animate the SVGs based on existing rated animations. AnimateSVG is, to the best of our knowledge, the first AI application that creates SVG animations from scratch, as a creative task that is usually entrusted to human designers. Understanding and learning from aesthetics ratings in the creation process has likewise seldomly been considered in extant research. Further, this is one of few studies considering SVG format as a data representation despite its significant importance in modern graphic design. Figure 1 depicts an exemplified animation of AnimateSVG.

Related Work

ML on Vector Graphics. Recent research on the application of ML methods to SVGs was successful in generating different elements, such as sketches and fonts (Ribeiro et al. 2020; Ha and Eck, 2018). However, often rasterized images are used in the generation and decoded into SVGs afterwards (Lopes et al. 2019). An essential study on generating SVGs directly is DeepSVG (Carlier et al. 2020). They find a meaningful embedding for vector graphics, which they apply to sample new vector graphics and interpolate them.



Figure 1. Exemplified Animation Created by the Proposed AnimateSVG Application.

Animating Vector Graphics using ML. ML is applied to create naturally looking, predefined character motions (Taylor et al. 2017), motion imitation, i.e., retargeting observed motions onto other shapes (Choi & Kim 2019), or animating transitions between graphics, e.g., through interpolation (Carlier et al. 2020). However, to the best of our knowledge, ML methods have not yet been applied to create entirely novel, aesthetic animations from an animation space. To apply ML for animating SVGs benefits from SVG’s native support of graphic animations¹.

Capturing Aesthetics with ML. With the advent of generative ML methods (e.g., Goodfellow et al. 2014; 2016) the question arises to which extent they can create artefacts that are perceived as aesthetically pleasing. Aesthetics refers to visual beauty, which has immediate and far-reaching effects on human perceptions (Maquet 1989; Lavie and Tractinsky 2004) and is thus essential for successful design. While there is much research dealing with explicating the rules of aesthetics and finding ways to statically evaluate it (Ngo et al. 2002; Tractinsky et al. 2000; 2006), such approaches are context-specific and cannot be applied to just any object in an observer-independent way. As a remedy, ML methods were experimented and found successful in capturing implicit and subjective phenomena like aesthetics from human-rated examples of a target class (Bansal and Bhattacharya 2013, Dhengre et al. 2020, Mateja and Heinzl 2021, Murray et al. 2012, Ramesh et al. 2022, Wang et al. 2016).

Data Acquisition and Preprocessing

To develop our envisioned AI animation application, there is no dataset available that contains animated SVGs in a standardized format. Thus, we acquire and preprocess a comprehensive dataset of logos and define a space of possible animations, which can be embedded in the SVGs.

Data and Data Preprocessing

To create a dataset for the purpose of this study, we source a set of 406 diverse SVG logos¹. As SVG representations of logos heavily vary depending on the designer’s style, the logos were preprocessed to a canonical form: All SVGs are scaled to a normalized viewbox. Commands were oriented

clockwise and the topmost, leftmost point was selected as the element’s starting position. All basic elements (rectangle, circle, ellipse, line, polyline and polygon) were converted to SVG paths, so that the resulting SVG logo $L = \{P_1, \dots, P_n\}$ is a set of n SVG paths P_i , describing all elements and their style features.

The logos are decomposed to path-level for the subsequent steps. A numerical representation for the normalized SVGs on path-level is obtained by applying the pretrained DeepSVG model (Carlier et al. 2020). DeepSVG represents and generates arbitrarily complex vector graphics. It encodes every shape separately into a tensor representation to create a latent vector thereof by relating the coded shapes to one another. An embedding $e_i \in \mathbb{R}^{256 \times 1}$ for each path P_i is obtained. To reduce this, we apply Principle Component Analysis (PCA) with a threshold to explain 99% of the variance in the observed data, yielding a reduced embedding $e_{red,i} \in \mathbb{R}^{10 \times 1}$. Depending on the designer, an SVG logo can consist of many small, possibly irrelevant element paths. Thus, an algorithm is implemented to derive the eight most important paths of an SVG logo. For each path, the mean-squared error (MSE) is computed between the original image (in PNG format) and the original image reduced by the respective path – the higher the MSE, the higher the relevance of said path for the logo. As DeepSVG neglects style information, which are essential for the task of aesthetically animating logos, we engineer 16 additional features to incorporate contextual style information about the entire logo: most common color of an SVG, color of the path at hand, difference between mean and path color, position and size of elements (implicitly also used in DeepSVG), as well as the number of paths in total. These additional features are used as a style vector $s_i \in \mathbb{R}^{16 \times 1}$ per path P_i . These style features are appended to our path-level embeddings. This SVG logo dataset was enhanced with human annotations for the purpose of animating the SVG logos in two ways:

- **Path Selection:** The eight most important paths of a logo are displayed and those paths, that should be animated, are selected by the annotators.
- **Path Animation Rating:** For every path in a logo, multiple randomly-generated animations are displayed and rated on a scale from 1 - very bad - to 5 - very good.

¹ We greatly appreciate the graphic design studio Bleil + Grimm GmbH for providing logos for the purpose of this study.

A handcrafted labelling website² was used to get a diverse perspective on aesthetics perception, yielding a total of ~ 15.000 annotations.

Animation Space

To enable the autonomous creation of new artefacts through computational methods, a structured conceptual space of possible solutions is required, which can be explored by the computational method (Boden 1998). In our study, we refer to this as the animation space. A logo animation arises from the combination of its animated paths, where each animation transforms certain path characteristics. To steer animations of vector graphics, SVG’s native animation language, called *Synchronized Multimedia Integration Language* (SMIL), allows for time-based manipulations of target paths. To construct the animation space, SMIL’s *animate* and *animateTransform* class are selected, where the former encompasses manipulations of style characteristics, while the latter provides manipulations for position and movement over time. We select six animation types from these two classes for our study, namely *translate*, *scale*, *rotate*, *skew*, *fill*, and *opacity*. Each animation is defined by a ‘from’ and a ‘to’ statement, i.e., the animation beginning and the final state of the logo after the animation. As the logos must return to their original state, the ‘to’ statement always encompasses the original logo. The parameters of the ‘from’ statement follow from the animation type, e.g., a *rotation* requires one parameter specifying its degree. A summary specifying the animation statements with pre-specified ranges for each possible parameter is provided in Table 1.

The 12-dimensional animation statement is constructed as follows: the animation types are one-hot encoded, i.e., each element can be associated with one animation type, and the appropriate parameters are specified. This is exemplified for a rotate animation in Figure 2. These animation statements

can be inserted into XML statements to animate the logo, using a handcrafted transfer function. Some additional rules are required to generate the XML statements:

- **Duration:** Each animation is 4 seconds long.
- **Begin:** The starting time of an animation is assigned from the left-most to the right-most animated element, every 0.5 seconds. Color animations start at 0 for values < 2. For values >= 2, two animation statements are set up, such that we initialize the characteristic with the original color and transform it over the most common color back to the original color. This way, color animations are smooth.
- **Fill:** The color is selected by determining the most common one (or if equal to the original color, the second most common color) in an SVG (neglecting black and white).

Animating SVG Logos

Our SVG logo animation application³ encompasses two steps: First, a path selection model is trained to determine suitable logo elements to be animated. Second, novel animations are created for the selected paths.

Step 1: Path Selection

In practice, not every element of an SVG logo should be animated to detain from overloaded animations. Thus, AnimateSVG decides which of the SVG paths to animate. A binary classifier is trained using ~ 3500 labeled logo paths. The 10-dimensional path embeddings together with the additional style features per logo, yielding a 26-dimensional vector per path, are used to predict which paths to animate in new logos. Multiple classification models are implemented and optimized, including logistic regression, random forest, and XGBoost, and finally an ExtraTrees classifier with 94.36% accuracy on a separate test set is selected.

TYPE	DESCRIPTION	PARAMETERS
<i>translate</i>	Moves an element from one point to another along a connecting line	$t \in [0,1]$ indicating the movement along the x- and y-axis when considered as a grid along which to move from left to right and top to bottom
<i>scale</i>	In- or decreases the size in relation to the original size	$s \in [0,1]$ if $s < 0.5$, the size of the element is increased from small to its initial size; else, the size is decreased from big to its initial size
<i>rotate</i>	Rotates an element from its original position over a specified rotation angle back to that position	$r \in [0,1]$ is scaled to a value between -360 and 360 to indicate the rotation angle
<i>skew</i>	Skews the element horizontally or vertically	$(sk_x, sk_y) \in [0,1]^2$. If $sk_x < 0.5$, the element is skewed along the x-axis; else, the element is skewed along the y-axis. sk_y determines the magnitude and ranges between -width/20 (height/20) and width/20 (-height/20)
<i>fill</i>	Changes the color over time	Set to the most common color
<i>opacity</i>	Changes the opacity over time	Set to 0

Table 1. Animation Space.

² Labeling Website: animate-logos.web.app

³ Implementation: github.com/AnimateSVG/AnimateSVG.git

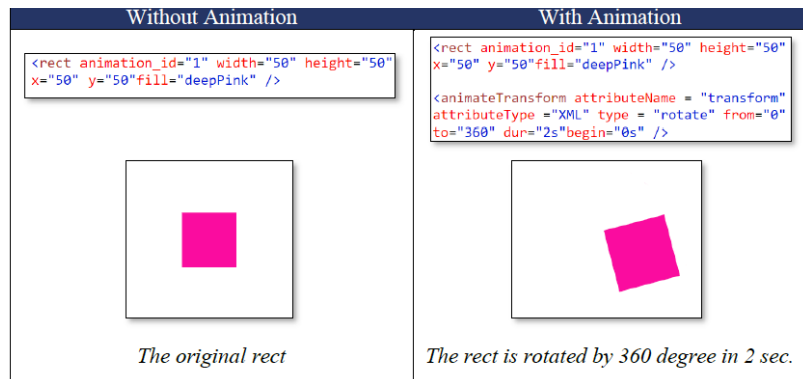
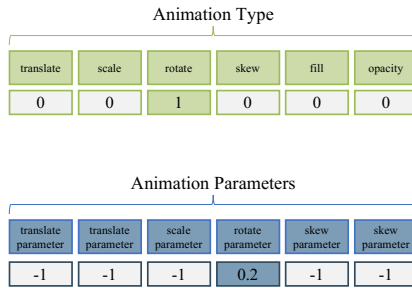


Figure 2. Animation Statement and XML Code Derived from Animation Statement.

Step 2: Logo Animation Generation

In the following, we present two alternative computational approaches to autonomously create logo animations.

Pretraining an Aesthetics Loss Function

The aesthetics of a logo animation can hardly be grasped with an explicit set of rules but rather emerges from the interplay of the animation with the composition of the logo. To capture these implicit aspects, we train an ML model to approximate human animation aesthetics to be used as an aesthetics loss function in the generation. The collected path-level animation ratings are used for training and each path is represented by its ten-dimensional path embedding, together with the 16 additional style features of the logo and the 12-dimensional numerical representation of path animations, yielding a 38-dimensional input space. This allows the loss model to predict the aesthetics quality of an animation on path-level in the context of the logo. A neural network with one 28-dimensional hidden layer and ReLU activation is set up. The model architecture is kept simple to accommodate for the limited training data. Aesthetics quality, as the target variable, is encoded on an ordinal scale. To facilitate that, an approach to encode an ordinal scale as a binary target vector for training neural networks is followed (Cao et al. 2020; Cheng, Wang, and Pollastri 2008): the Consistent Rank Logits (CORAL) framework guarantees rank-monotonicity. The rating categories are extended into binary variables, transforming the learning problem into four binary classification problems, which share the same weight parameters but have independent bias units. The summed cross-entropy of the binary classifiers is minimized using Adam (Kingma and Ba 2015).

With the optimized and fixed parameters of the aesthetics evaluation model in place, a generator network is proposed. It creates animations from scratch and gets feedback from the pretrained aesthetics evaluation model during training. The generator applies a discriminative learning approach

that takes as input the logo as context of each path to meaningfully animate it. The 26-dimensional combination of the path embedding and style features of the logo are used as an input structure to determine the animation type as a first step. This way, each path can be animated in the context of the logo, even if the logo itself has been unseen by the animation generation network. Using a fully connected 15-dimensional hidden layer with ReLU activation, this input is transformed into a 6-element interim output layer, which returns the probabilities for each animation type after applying softmax function. The output vector of this layer is concatenated with the original input vector, such that a 32-element representation of the paths is obtained, which holds information about the predicted animation type. This vector is used as input to a 20-dimensional fully connected hidden layer to determine animation parameters. Again, ReLU activation is applied before going over to a 6-element output layer encompassing the animation parameters. The output of this discriminative generation model is the 12-element concatenation of the two interim output layers, where the first six elements capture the animation type and the last six elements capture animation parameters as shown in Figure 3.

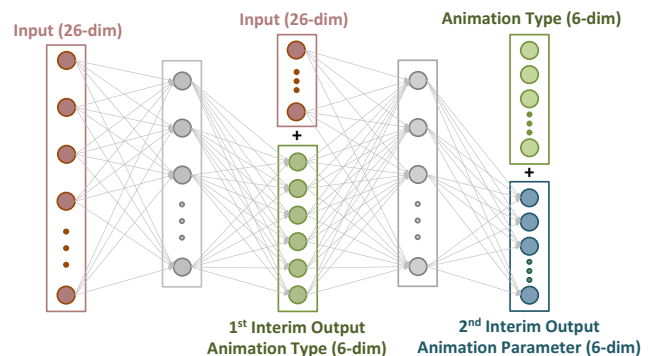


Figure 3. Dependent Generation of Animation Types and Parameters in Neural Network Trained with an Aesthetics Loss.

Tree Ensembles with Global Optimization

Many ML applications yet suffer from limited data availability. While we obtained successful results with our own dataset, oftentimes neither time nor resources are available for such costly data acquisition. To bridge this gap, we implement a second computational method for logo animation that significantly reduces the demands on data availability. It combines gradient boosted trees and other regression tree models in tree ensembles with global optimization. We use the ENTMOOT (ENsemble Tree Model Optimization Tool) framework (Thebelt et al. 2021), which leverages the strengths of tree models to effectively manage sparse data and their excellent prediction capabilities. It incorporates an uncertainty measure that is compatible with tree models to solve larger optimization problems in an uncertainty-aware manner, which tree models alone fail to do. Specifically, ENTMOOT encodes tree-based machine learning models, uncertainty estimates, and applicable constraints as a mathematical problem and uses deterministic global optimization to solve it. We select a representative subset of 100 logo animations with their ratings as input samples. ENTMOOT then performs the following steps:

- *Tree ensemble learning*: First, a tree ensemble model is trained for approximating an animation evaluation function. These tree models effectively manage sparse data and categorical or conditional feature spaces, and do not require variable scaling. They work well as interpolators, i.e., they have good prediction performance within the training data. A random forest is used for the tree ensemble model as it outperformed gradient boosting classifiers.
- *Uncertainty estimation*: For unexplored regions of the input space not covered by the training data, tree-based models have decreasing prediction performance (Thebelt et al. 2021). Thus, an uncertainty measure u is introduced, which penalizes data points that lie far away from the training data D , and is defined as $u(x) = \min_{x_d \in D} \|x_d - x\|_1$, where $\|\cdot\|_1$ denotes the $L1$ -norm.
- *Acquisition formulation*: Both the random forest model prediction \hat{y} and the uncertainty measure u are combined in an acquisition function. Animations are desired that maximize the model’s predicted rating. However, as tree-based models have decreasing prediction performance in unexplored regions, the uncertainty term is weighted with a hyperparameter κ , which we set to 1.96 similar to Thebelt et al. (2021), resulting in an acquisition formulation of $\min_x - \hat{y} + 1.96 * u(x)$. In our optimization problem, the objective is represented by this acquisition function and is fixed after the training phase.

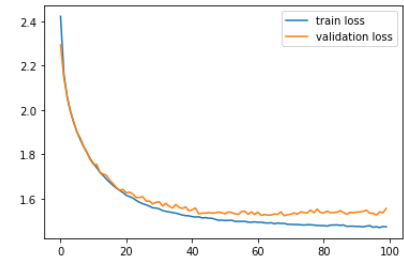
At inference, a mixed-integer problem is solved in a deterministic manner for each path individually, with the trained acquisition as a common objective function. The constraints to the optimization problem are defined by the respective path itself: Given a certain path, its numerical representation is extracted and fixed in the input space. Thus, only the vector that defines the animation is free to be optimized and

chosen in the process, where further constraints are implemented to ensure that it has a feasible format. The resulting highly complex mixed-integer problems were solved using Gurobi, a state-of-the-art global optimization software.

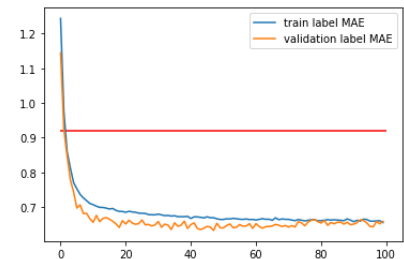
Results and Discussion

For step 1, the path selection, we reported a test accuracy of 94.36%. For step 2, the animation generation, two alternative approaches were suggested. For the aesthetics generation network, we first consider the aesthetics evaluation model. CORAL loss (Cao et al. 2020) as an indicator for training success is reported in Figure 3. Train and validation loss steadily decrease and saturate eventually. Figure 4 further reports label mean absolute error (MAE) as an evaluation measure in comparison to a baseline majority classifier indicated by the red lines. We achieve an MAE of 0.8149 on test data (compare: 0.6977 on validation data), i.e., we can approximate human aesthetics evaluations successfully. The generator loss is determined using the aesthetics evaluation model. Figure 4 displays the train and validation loss of the animation generator. Early stopping is applied after epoch 30 as validation and train loss only minimally improve beyond this point, but a severe loss in the diversity of the created animations was observed. Indeed, the model started to

CORAL Loss of Aesthetics Evaluation Model



Mean Absolute Error of Aesthetics Evaluation Model



Aesthetics Loss of Animation Generation Model

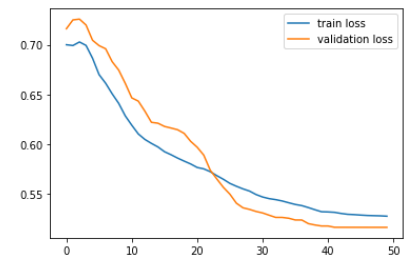


Figure 4. Performance of Aesthetics Generator and Evaluation.

overfit to a few animation types, which often receive good ratings from the aesthetics evaluation model. To retain the diversity in the animations, early stopping was necessary.

To compare the proposed computational methods, human subject evaluations build ground to evaluate the results and account for the observer-dependent nature of aesthetics. A separate hold-out test set is used to generate SVG animations with our two proposed computational approaches. ~1900 logo animations are rated by independent observers along a 1-5 ordinal scale. The animation generator network with a pretrained aesthetics loss, achieves an average rating of 3.6. This is slightly improved by the tree ensemble with global optimization, i.e., ENTMOOT, to an average rating of 3.7, i.e., both model’s logo animations receive ratings of ‘good’. There is no major malfunctioning observable as only 1% of the animated paths received a rating of ‘very bad’.

To demonstrate the effectiveness of our proposed architecture, an ablation study is conducted for three core components of our approach: (1) *Without path selection*, the model can predict animation types and parameters but ignores whether a path should be animated at all. Thus, all paths are animated, resulting in cluttered animations; (2) *Without animation type prediction*, the model predicts parameters for a randomly sampled animation type. While parameters can be found to match the given types, the missing alignment of types and logos results in reduced aesthetic quality; (3) *Without animation parameter prediction*, the model assigns random parameters after the prediction of the animation type. The model receives ambiguous evaluations as feedback, impeding the learning process altogether. We rerun our user study on a smaller sample (30 logos for each of the left-out components, annotated by 8 users) for each ablation. From this, we find every component of our architecture to be essential in the process of animating SVGs as depicted in Table 2. All ablations perform significantly worse than the full models in our AnimateSVG prototype.

Examples of our generated animations can be found in the provided Git repository. We conclude that our proposed AnimateSVG application can indeed generate aesthetic logo animations. It can meaningfully animate multiple paths in the context of the logo.

ABLATION		RATING
<i>(1) No path selection</i>		2.2
<i>(2) No animation type prediction</i>		2.6
<i>(3) No animation parameter prediction</i>		2.5
<i>AnimateSVG</i>	ENTMOOT	Aesthetics Generator
<i>(Full Model)</i>	3.7*	3.6**

(Paired t-test: AnimateSVG significantly outperforms all ablations with *p-value 0.0089 and **p-value 0.0104)

Table 2. Results of Human Subject Ablation Study.

Path Forward for Deployment

This work assesses the potential of applying AI for SVG logo animation. We developed an end-to-end solution that generates animations from SVG data, using human-perceived aesthetics as the primary target. To the best of our knowledge, we are the first to propose an AI application that comes up with entirely new SVG animations from a conceptual animation space. From a technical perspective, we achieve this by employing a novel network architecture that allows a dependent prediction of animation types and parameters. Despite SVG being the standard in modern graphic design, our study is one of the few to enable ML methods to use SVG and, hereby, interact with the systems that the world is actually made of.

We provide the application as a web service⁴, which enables users to upload an SVG logo and receive its animation. Both graphic design amateurs and experts can use the created animations to optionally postprocess and eventually embed them in their digital content. As a path forward in deployment, we envision to increase the compositional harmony, diversity, and resultingly the aesthetics in the created SVG animations. We believe that we can generate more harmonious animations by considering the overall logo animation rather than evaluating each path individually. For the diversity and aesthetics of the animations, the two proposed approaches go hand in hand in facilitating this: By leveraging the different demands on data and resulting animation characteristics, the two computational approaches can be used in different phases of deployment: The tree ensembles work with sparse data while still generating satisfying results, hereby, offering a solid starting point to reliably provide users with ‘good’ animations. However, ENTMOOT inherently limits the variety of animations as it interpolates within 100 data points, which cannot capture a broad variety in animation alternatives. Instead, the data-intensive aesthetics generator network can be continuously enhanced by probing new animation types and parameters and receiving immediate user feedback via the application. This way, users can be sure to at least receive a low diversity but satisfying animation and we can continue to collect labels for generated animations and enhance the animation space. To facilitate this continuous learning approach of the generator network, the application contains the possibility for users to rate the generated animations. These ratings can eventually be integrated into subsequent training processes. As a result, we expect a significant increase in the variety of animation types and the aesthetics of the animations. We consider this staged deployment approach with continuous feedback loops leverageable to other generative scenarios with a fixable conceptual space, that are often restricted by limited data availability.

⁴ Link to Application: animate-svgs.com

References

- Bansal, D. and Bhattacharya, S. 2013. Semi-supervised Learning Based Aesthetic Classifier for Short Animations Embedded in Web Pages. In *Human-Computer Interaction 8117*. Cape Town: Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-642-40483-2_51
- Boden, M. A. 1998. Creativity and Artificial Intelligence. *Artificial Intelligence* 103(1-2):347–356. doi.org/10.1016/S0004-3702(98)00055-1
- Bruce, N. I.; Murthi, B. P.; and Rao, R. C. 2017. A Dynamic Model for Digital Advertising: The Effects of Creative Format, Message Content, and Targeting on Engagement. *Journal of Marketing Research* 54(2):202–218. doi.org/10.1509/jmr.14.0117
- Cao, W.; Mirjalili, V.; and Raschka, S. 2020. Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation. *Pattern Recognition Letters* 140:325–331. doi.org/10.1016/j.patrec.2020.11.008
- Carlier, A.; Danelljan, M.; Alahi, A.; and Timofte, R. 2020. DeepSVG: A Hierarchical Generative Network for Vector Graphics Animation. In *Advances in Neural Information Processing Systems 33*. Virtual.
- Cheng, J.; Wang, Z.; and Pollastri, G. 2008. A Neural Network Approach to Ordinal Regression. In *Proceedings of the International Joint Conference on Neural Networks*. Hong Kong: IEEE.
- Choi, S. and Kim, J. 2019. Towards a Natural Motion Generator: A Pipeline to Control a Humanoid based on Motion Data. In *IEEE International Conference on Intelligent Robots and Systems*. Macau: IEEE.
- Dhengre, S.; Mathur, J.; Oghazian, F.; Tan, X.; and McComb, C. 2020. Towards Enhanced Creativity in Interface Design through Automated Usability Evaluation. In *Proceedings of the 11th International Conference on Computational Creativity*. Coimbra: Association for Computational Creativity.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 27*. Montréal.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. Cambridge: MIT Press.
- Ha, D. and Eck, D. 2018. A Neural Representation of Sketch Drawings. In *6th International Conference on Learning Representations*. San Juan.
- Kingma, D. P. and Ba, J. L. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*. San Diego.
- Lavie, T. and Tractinsky, N. 2004. Assessing Dimensions of Perceived Visual Aesthetics of Web Sites. *International Journal of Human Computer Studies* 60(3):269–298. doi.org/10.1016/j.ijhcs.2003.09.002
- Lopes, R. G.; Ha, D.; Eck, D.; and Shlens, J. 2019. A Learned Representation for Scalable Vector Graphics. In *Proceedings of the IEEE International Conference on Computer Vision*. Seoul: IEEE/CVF.
- Maquet, J. 1989. The Aesthetic Experience: An Anthropologist Looks at the Visual Arts. *The Journal of Aesthetics and Art Criticism* 47(2):200–200. doi.org/10.1086/adx.7.3.27947943
- Mateja, D. and Heinzl, A. 2021. Towards Machine Learning as an Enabler of Computational Creativity. *IEEE Transactions on Artificial Intelligence* 2(6):460–475, 2021. doi.org/10.1109/TAI.2021.3100456
- Murray, N.; Marchesotti, L.; and Perronnin, F. 2012. AVA: A Large-scale Database for Aesthetic Visual Analysis. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Providence: IEEE/CVF.
- Ngo, D. C. L.; Teo, L. S.; and Byrne, J. G. 2002. Evaluating Interface Esthetics. *Knowledge and Information Systems* 4(1):46–79. doi.org/10.1007/s10115-002-8193-6
- Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-conditional Image Generation with CLIP Latents. arXiv preprint. arXiv:2204.06125
- Ribeiro, L. S. F.; Bui, T.; Collomosse, J.; and Ponti, M. 2020. Sketchformer: Transformer-based Representation for Sketched Structure. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Seattle: IEEE/CVF.
- Taylor, S.; Kim, T.; Yue, Y.; Krahe, M. M.; Rodriguez, A. G.; Hodgins, J.; and Matthews, I. 2017. A Deep Learning Approach for Generalized Speech Animation. *ACM Transactions on Graphics* 36(4):1–11. doi.org/10.1145/3072959.3073699
- Thebelt, A.; Kronqvist, J.; Mistry, M.; Lee, R. M.; Sudermann-Merx, N.; and Misener, R. 2021. ENTMOOT: A Framework for Optimization over Ensemble Tree Models. *Computers and Chemical Engineering* 151:1–33. doi.org/10.1016/j.compchemeng.2021.107343
- Tractinsky, N., Katz, A., and Ikar, D. 2000. What is Beautiful is Usable. *Interacting with Computers* 13(2):127–145. doi.org/10.1016/S0953-5438(00)00031-X
- Tractinsky, N.; Cokhavi, A.; Kirschenbaum, M.; and Sharfi, T. 2006. Evaluating the Consistency of Immediate Aesthetic Perceptions of Web Pages. *International Journal of Human Computer Studies* 64(11):1071–1083. doi.org/10.1016/j.ijhcs.2006.06.009
- Wang, W.; Cai, D.; Wang, L.; Huang, Q.; Xu, X.; and Li, X. 2016. Synthesized Computational Aesthetic Evaluation of Photos. *Neurocomputing* 172:244–252. doi.org/10.1016/j.neucom.2014.12.106
- Walsh, M.F.; Page Winterich, K.; and Mittal, V. 2010. Do Logo Redesigns Help or Hurt your Brand? The Role of Brand Commitment. *Journal of Product & Brand Management* 19(2):76–84. doi.org/10.1108/10610421011033421

ⁱ W3C. 2011. Scalable Vector Graphics (SVG) 1.1 (Second Edition). w3.org/TR/SVG11/. Accessed: 2022-11-14.