Stroke Patches: Customizable Artistic Image Styling Using Regression

Ian Jaffray
ian.jaffray@outlook.com

John Bronskill
University of Cambridge
jfb54@cam.ac.uk

Abstract

We present a novel, regression-based method for artistically styling images. Unlike recent neural style transfer or diffusion-based approaches, our method allows for explicit control over the stroke composition and level of detail in the rendered image through the use of an extensible set of *stroke patches*. The stroke patch sets are procedurally generated by small programs that control the shape, size, orientation, density, color, and noise level of the strokes in the individual patches. Once trained on a set of stroke patches, a U-Net based regression model can render any input image in a variety of distinct, evocative and customizable styles.

1 Introduction

Transforming images so that they appear to be rendered in an artistic style has a long history. Gallery Effects [1–4] employed a graph of image processing operators to render images as watercolor, charcoal, and pen and ink, among others. Haeberli [5] created impressionist renderings by specifying the position, color, size, direction, and shape of each stroke driven by features of the source image. Gatys et al. [6] introduced the concept of neural style transfer by using neural networks to separate the content and style of an image, allowing the style from a source image to be imparted on a target image while maintaining the content. More recently, diffusion based methods [7, 8] and their implementation in various products [9–11] enables the styling of an input image by describing the desired artistic appearance using a natural language prompt.

While the recent machine learning-based approaches excel at capturing the overall *feel* of a specified style, they lack control over the composition of the strokes and the amount of detail in the rendered images. We propose a complementary approach to artistically styling images that allows explicit and extensible control over image detail through the use of procedurally generated *stroke patches*. By controlling the shape, size, orientation, density, color, and noise level of the strokes in the stroke patches, a wide variety of expressive artistic looks can be achieved.

Our contributions are twofold. The first is the concept of a customizable, procedurally generated stroke patch, which defines an artistic look and the level of detail in the stylized output. The second is using a regression-based approach that, once trained on a set of stroke patches, learns to map continuous tone regions of an image to the discrete strokes specified in the stroke patches.

Section 2 defines a stroke patch and how to create them, the model architecture, and the details of training and inference. Section 3 depicts a range of outputs from the method and their associated stroke patches, as well as the effect of varying certain stroke patch parameters. Finally, Section 4 summarizes our work and places it in perspective with existing artistic image rendering methodologies.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

2 Method

In this section, we define the notion of a set of stroke patches, the overall system architecture, and detail training and inference.

Figure 1: Example stroke patches.

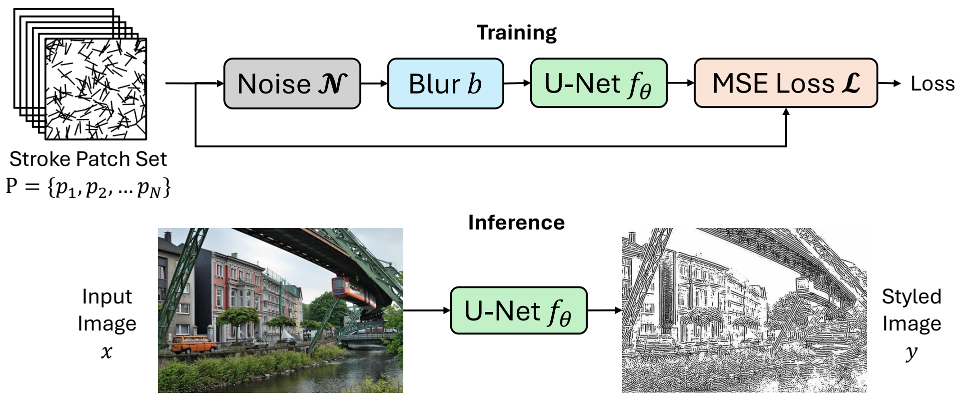

Figure 2: **Training (top)** The training set is a set of stroke patches P. Each stroke patch p_n has noise $\mathcal{N}$ added and then is blurred with a Gaussian kernel b and input to U-Net f_θ. The output of these steps is then used to compute the MSE loss with respect to p_n. **Inference (bottom)** An arbitrary input image x is passed through the trained U-Net f_θ yielding a stylized image y.

Stroke Patches Figure 1 shows a single stroke patch extracted from five different stroke patch sets. We create a set of related stroke patch images procedurally by randomly varying various patch parameters such as stroke shape, size, location, orientation, color, and noise level. An algorithm for generating a set of stroke patches for the Wet Brush style is specified in Algorithm 1.

Architecture Our approach is depicted in Figure 2. The main component is a U-Net [12] f with parameters θ. We make two minor changes to the standard U-Net configuration. The first is that we place a sigmoid operator at the output to ensure that the output values always lie in the 0 to 1 range. The second is that we have replaced all batch normalization operators [13] with instance norm operators [14] allowing us to train on relatively large stroke patches with small batch sizes on modest GPUs. During training, we sometimes add noise $\mathcal{N}$ (usually Gaussian or uniform) to the patches and always use a Gaussian blur operator b (typically with radius 5.0) to soften the stroke patch images.

Training and Inference To train f_θ, we minimize the following mean-squared error (MSE) loss:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(b(p_i + \mathcal{N})) - p_i)^2 \tag{1}$$

At inference, to create an artistic image y, we pass an input image x through the trained f_θ as follows:

$$y = f_\theta(x). \tag{2}$$

During training, the model f_θ learns a mapping from the blurred and noised stroke patches to the unprocessed stroke patches. During inference, the model transforms continuous tone areas in the input image that are somewhat similar to a stroke in a stroke patch into a clean, well-defined stroke in the stylized output.

Algorithm 1 Wet Brush stroke patch set generation

Require: W, H: stroke patch width and height in pixels (default = 400)
Require: N: number of stroke patches in stroke patch set (default = 5000)
Require: BG: background color (default = 'white'); L: stroke length (default = 80)
Require: S: number of strokes in each patch (default = 50); T: stroke thickness (default = 40)

procedure CREATEWETBRUSHSTROKEPATHSET(W, H, N, BG, S, T)
  stroke_patch_set ← [ ] ▷ set of stroke patches
  for $i \leftarrow 1$ to N **do** ▷ create N patches
    patch ← `create_image`(W, H, BG)
    for $i \leftarrow 1$ to S **do** ▷ create S strokes in each patch
      $\phi \leftarrow \mathcal{U}(0, 2\pi)$ ▷ line orientation
      $x_1 \leftarrow \mathcal{U}(-L/2, W + L/2), y_1 \leftarrow \mathcal{U}(-L/2, H + L/2)$ ▷ line start
      $x_2 = x_1 + L\cos(\phi), y_2 = y_1 + L\sin(\phi)$ ▷ line end
      $C \leftarrow$ `SetRGBAColor`($\mathcal{U}(0,1)$,$\mathcal{U}(0,1)$,$\mathcal{U}(0,1), 1$) ▷ Random opaque RGB color.
      patch.`DrawLine`($x_1, y_1, x_2, y_2, L, T, C$, `round_end_cap`)
    end for
    stroke_patch_set.append(patch)
  end for
  `return`(stroke_patch_set)
end procedure
$\mathcal{N}$ is Gaussian with $\mu = 0$ and $\sigma = 500$. Gaussian blur b radius is set to 5.0.

3 Results

All training and inference was performed on a Nvidia 4090 GPU. Unless otherwise stated, we train with a set P of N=5000 stroke patches of size 400 × 400 pixels for 10 epochs using the Adam optimizer [15] with a learning rate of 0.001, and a Gaussian blur (b) radius of 5.0. Training takes 98 seconds per epoch, and inference takes less than a second. Source code is available at `https://github.com/jfb54/stroke_patches`.

Figures 3 and 4 show three original unprocessed images and stylized results when trained on five different sets of stroke patches. Despite the simplicity of the procedurally generated stroke patches, the method yields a wide variety of distinctive artistic effects. Figure 6 shows how stylized results can be made more prominent by reducing the size of an original image by a fixed fraction r, applying the model, and then enlarging the resulting image back to its original size. More precisely, the output is $y = R_{1/r}(f_\theta(R_r(x)))$, where R_r indicates the spatial image scaling operator by a factor $0 < r \leq 1$. Figure 5 shows the effect of modifying various stroke patch set generation parameters including the number of the strokes in each patch, the stroke width and length, the amount of Gaussian noise added, and the stroke opacity. The specification of these parameters allows for a significant degree of control over the result.

4 Summary, Limitations, and Societal Impact

Summary In this work, we presented a regression-based method to artistically render an image that allows for explicit control over the stroke composition and level of detail in the output through the use of an extensible set of stroke patches.

Limitations The primary limitation of this work is that, unlike style transfer or diffusion methods, the overall high-level look of the image cannot be specified by a text prompt or reference to another image. Our work requires the design and specification of low-level detail via a set of stroke patches. Our approach does not replace existing style transfer, diffusion-based, or other effect methods, but instead complements them, giving the digital artist a new customizable tool with a high degree of control to transform images or video to have an artistic or painterly appearance.

Societal Impact We hope that our work will be useful for both amateur and professional artists to create artistic imagery, perhaps saving time and money when compared to hiring an artist who works in traditional media. We see no negative societal impact as our model is limited in capability and it would not be possible to use the model to generate counterfeit works.

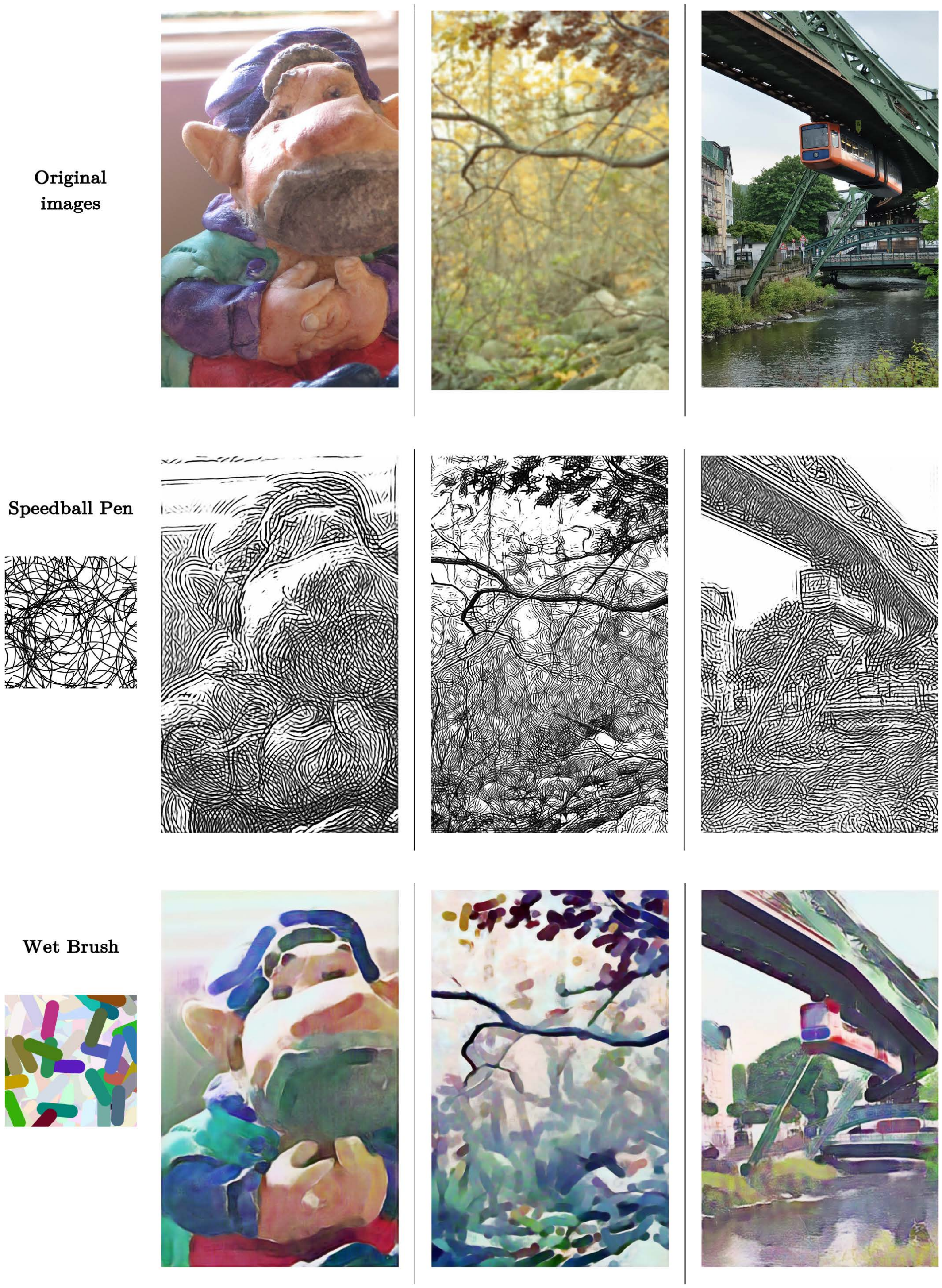

Figure 3: **Results Part 1.** The top row depicts three original unprocessed images and the middle and bottom rows show the results of applying the model trained on two different stroke patches sets (Speedball Pen and Wet Brush). A sample stroke patch is shown to the left of the stylized images. **Original photo credits:** Ian Jaffray (left, center), Mbdortmund, GFDL 1.2 `http://www.gnu.org/licenses/old-licenses/fdl-1.2.html`, via Wikimedia Commons `https://commons.wikimedia.org/wiki/File:Wuppertal-100508-12825-Uferstra%C3%9Fe.jpg` (right).

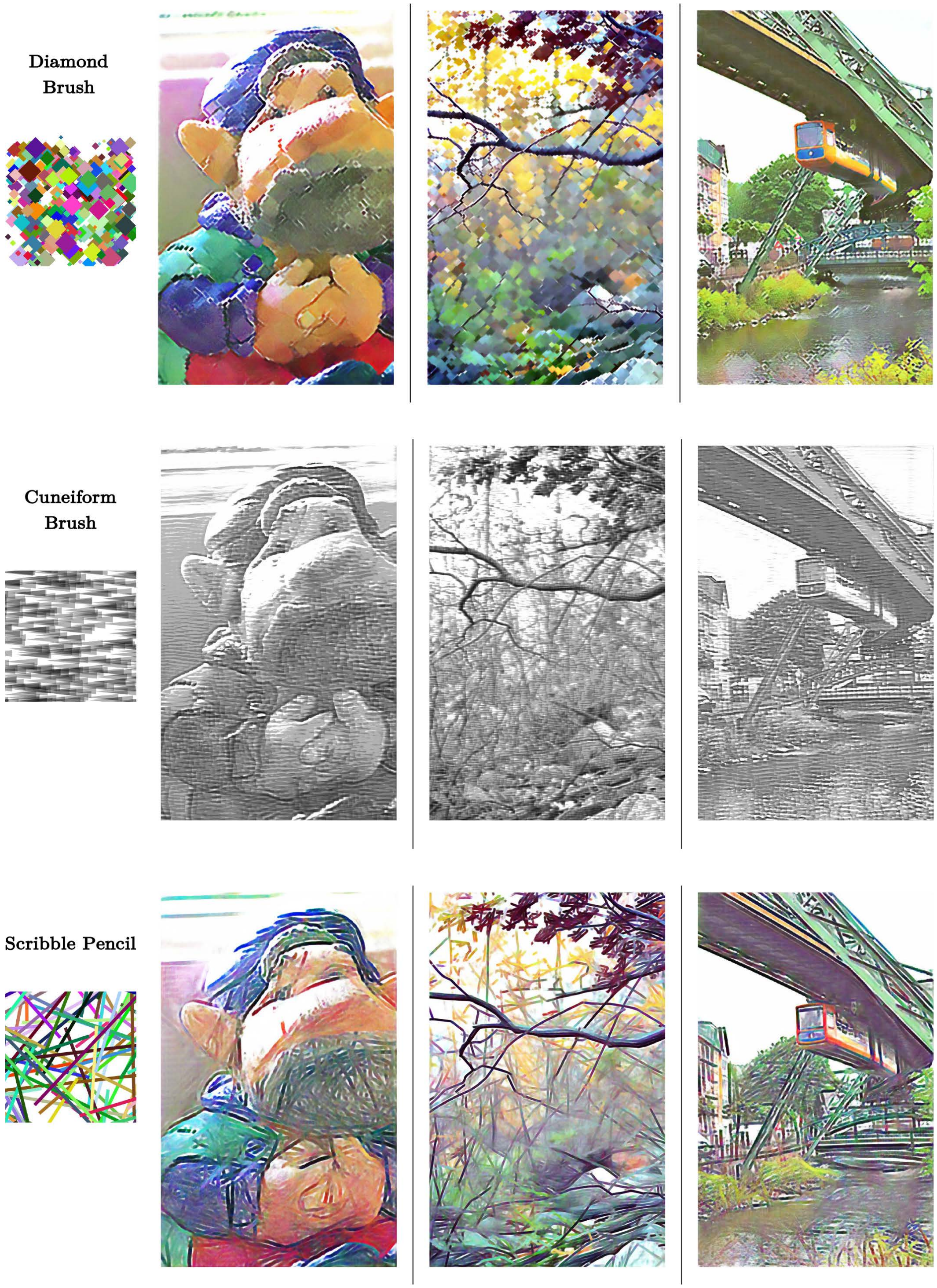

Figure 4: **Results Part 2.** Each row shows the results of applying the model trained on three different stroke patches sets (Diamond Brush, Cuneiform Brush, and Scribble Pencil) using the same original images as Figure 3. A sample stroke patch is shown to the left of the stylized images.

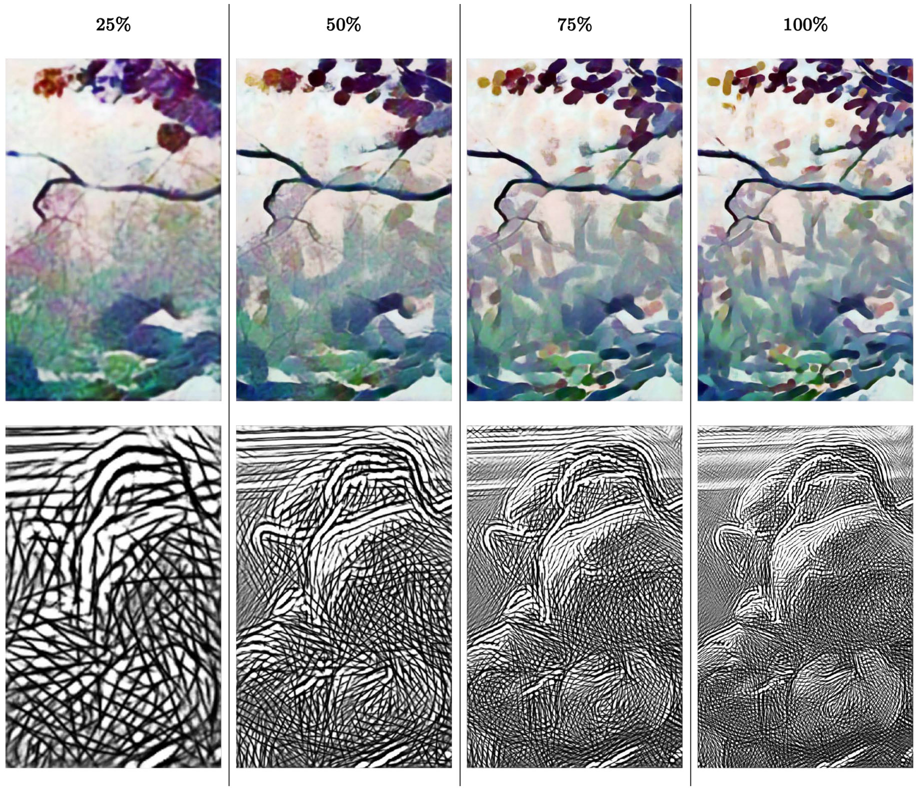

Figure 5: **Effect of scaling the original images.** Each column shows the effect of reducing the original image size by the percentage specified in the column heading before applying the model trained on the Wet Brush (top row) and Speedball Pen (bottom row) stroke patch sets and then enlarging the resulting image back to its original size.

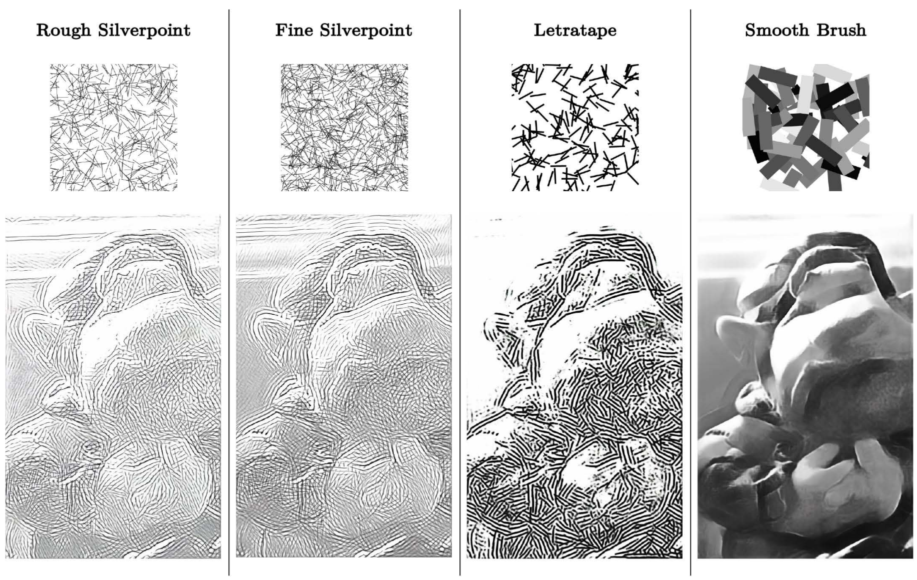

Figure 6: **Effect of changing stroke patch set generation parameters.** The Rough Silverpoint style uses 700 black strokes that are 50 pixels long and 1 pixel wide. The Fine Silverpoint style increases the number of stokes to 1200. The Letratape style decreases the number of strokes to 200, but increases their width to 5 pixels. Finally, the Smooth Brush style reduces the number of strokes to 50, but increases the stroke length to 120 pixels and the stroke width to 40 pixels and adds Gaussian noise with μ=0 and σ=500.

References

[1] Wikipedia Contributors. Photoshop plugin. *Wikipedia*, 2025. URL `https://en.wikipedia.org/wiki/Photoshop_plugin`. [Online; accessed 2025-06-29].

[2] Jaffray, Ian and Bronskill, John. Apparatus and method for transforming a digitized signal of an image. US Patent 5063448, November 5, 1991, 1991. URL `https://patentimages.storage.googleapis.com/64/bd/a8/f54dc54674eff9/US5063448.pdf`. United States.

[3] Jaffray, Ian and Bronskill, John. Apparatus and method for transforming a digitized signal of an image to incorporate an airbrush effect. US Patent 5245432, September 14, 1993, 1993. URL `https://patentimages.storage.googleapis.com/ce/63/31/ef4ba1490ebe10/US5245432.pdf`. United States.

[4] Jaffray, Ian and Bronskill, John. Apparatus and method for transforming a digitized signal of an image into a reflective surface. US Patent 5325200, June 28, 1994, 1994. URL `https://patentimages.storage.googleapis.com/cc/6c/dd/8331cc4b5b8c5f/US5325200.pdf`. United States.

[5] Paul Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, 1990.

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

[7] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/sohl-dickstein15.html`.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[10] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[11] Google. Gemini 2.5, 2025. URL `https://gemini.google.com/`. Accessed: 2025-07-02.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[14] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim three contributions in our paper in Section 1 and each is described in detail in Section 2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to Section 4 where we discuss limitations of our work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We supply all experimental setting/details required to reproduce the experiments. We do this by supplying full source code as well as thorough information in Sections 2 and 3 and Algorithm 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide source code to reproduce the results at `https://github.com/jfb54/stroke_patches`. Along with the code, we provide a README file that details installation, configuration, and options in order to reproduce the results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We supply all experimental setting/details required to reproduce the results. We do this by supplying full source code as well as thorough information in Sections 2 and 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experimental results contain only image outputs. No numerical results are stated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3 details compute resources used and processing times needed to reproduce the experiments. The full research project required more compute than the experiments reported in the paper due to early and failed experiments that didn't make it into the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and have complied with them.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Refer to Section 4 for a discussion of positive and negative societal impact of our work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The models that we will release are low-risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In our paper we use the following assets:

- We properly credit the original photos that we use for styling in a caption where they are first used.
- We properly credit and respect the license for the open-source code to implemented the U-Net in our source code.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the following assets, which are fully documented and licensed in our source code repository:

    - Source code for reproducing the results.
    - Pretrained models for the styles used in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This work did not involve any use of LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.