FontCLIPstyler: 言語によるシーンテキストスタイル変換

原 虹暉[1,a)]　柳井 啓司[1,b)]

概要

シーンテキスト編集は、ポスターデザインなどの分野で広く使用されている。既存の研究では、テキストのスタイルと画像背景を変更せずに画像内のテキスト内容を変更することは実現されているが、画像のテキスト領域のスタイルを自由に変更することがまだ実現されていない。そこで本研究では、CLIPstyler [12] をベースに、シーンテキストのスタイル変換を実現する新たなフレームワーク FontCLIPstyler を提案する。提案手法は、プロンプトを用いて、画像内のテキストを任意のスタイルに変換することに成功した。実験結果は、提案手法が画像の背景とテキストの内容を変更することなく自然なスタイル付きのシーンテキストを生成できることを示している。

1. はじめに

近年、ディープラーニングの発展に伴い、画像編集がますます便利になり、シーンテキスト編集も注目されている。これまでのシーンテキスト編集手法は、シーン画像内のテキストを他のテキストに置き換え、背景とテキストスタイル（色、テクスチャなど）を変更しないことを実現した。しかし、これらの手法では、テキスト内容の置き換えのみを注目しており、テキストのスタイルを自由に変更することはできない。したがって、本研究では図 1 のように、画像内のテキスト内容と背景を変更することなく、テキストのスタイルのみを変更するシーンテキストスタイル変換タスクを提案する。既存のシーンテキスト編集手法は通常、背景の Inpainting、テキスト変換と画像再合成 3 つのステップで行う。このため、テキスト変換する際に、生成されたテキストが元テキストと同じスタイルになるために、元画像をスタイル参照画像として利用する必要がある。最新の拡散モデルベースの手法は、より自然なシーンテキスト画像が生成できるが、テキストを編集する際に、テキストのスタイルが画像内にある他のテキストのスタイルを参考に生成され、自由に変更することができない。

そこで本研究では、シーンテキストのスタイル変換を実現するために、CLIPstyler [12] ベースの新しいフレームワーク FontCLIPstyler を提案する。提案手法は、参照スタイル画像を必要とせず、画像の背景とテキストの内容を変更することなく、プロンプトを用いて画像内のテキストを任意のスタイルに変換することを実現した。

図 1　本研究と CLIPstyler のシーンテキストスタイル変換の結果

2. 関連研究

シーンテキスト編集は、テキストの外観を維持しながら、元の画像内のテキストを別のテキストに置き換えることが大きな進歩を遂げた。STEFANN [19] は、フォントの構造と色の変換をそれぞれ 2 つのネットワークを設計し、一文字ずつでテキストを置き換える。しかし、元と違う文字数にすることはできない。SRNet [23] は、背景復元、テキスト変換、再合成の 3 つのサブネットワークを利用してテキストを置換する。SwapText [24] は SRNet をベースに、TPS(Thin-Plate-Spline) モジュールを導入し、空間点を使用してテキストを幾何学的に変換する。SimAN [13] は類似性認識の正規化を導入し、自己教師あり学習法でネットワークを学習する。TextStyleBrush [11] は StyleGAN [10] に基づき、テキスト画像のスタイルベクターを Generator に導入、最終画像の生成をガイドする。Mostel [15] は追加のストロークレベル情報を導入し、合成データと実世界データの併用をすることで、シーンテキスト編集のパフォーマンスを大幅に向上させる。しかし、これらの手法はスタイル参照画像が必要である。Diffusion model [17] は画像編集において大きな成功を収めており、DiffSTE [8]、DiffUTE [2]、GlyphDraw [14]、GlyphControl [25]、TextDiffuser [3] などの手法は、拡散モデルを使用して自然なシーンテキスト生成と編集を実現している。しかし、テキストのスタイル

1　電気通信大学
a)　yuan-h@mm.inf.uec.ac.jp
b)　yanai@cs.uec.ac.jp

をコントロールすることができない。本研究は、スタイル参照画面が必要なく、プロンプトでシーンテキストのスタイルを指定することができる。

画像スタイル変換は、参照画像のスタイルをターゲット画像に転送することを目的としている。StyleGAN [10]、StyTr2 [4] など多くの手法は GAN [5] と Transformer [22] を使用することで、スタイル変換において大きな成功を収めている。しかし、これらの手法はスタイル参照画像を必要とする。最近の CLIPstyler は、図 2 のようにプロンプトを用いて任意のスタイル変換を可能にすることで、この問題を改善した。Sem-CS [9] と Gen-Art [26] は、semantic segmentation を使って、CLIPstyler の画像の前景部分の over-stylization 問題を解決した。しかし、これらの手法は画像全体的にスタイル変換を行い、画像内の特定のターゲットに対するのスタイル変換を実現することができない。Word as Image [6]、CLIPFont [20]、DS-Fusion [21]、Zero-shot Font Style Transfer [7] などの手法はプロンプトでフォントのスタイル変換を実現したが、これらの手法はフォント画像しか対応できない。本研究では、提案する FontCLIPstyler を利用することで、プロンプトでシーン画像内のテキスト領域のスタイル変換を実現する。

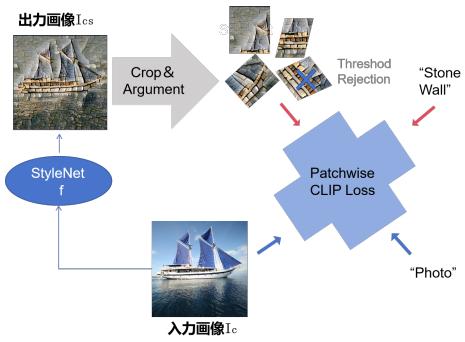

図 2 CLIPstyler 概要図

3. 手法

本研究では、シーン画像内のテキスト領域のスタイル変換を実現するため、CLIPstyler をベースにする新たな FontCLIPstyler を提案する。提案手法の概要は図 3 に示す。提案ネットワークは主にシーンテキスト画像のテキスト部分の Mask 画像を抽出する m(MaskNet) ネットワークとスタイル変換を行う CNN encoder-decoder ネットワーク f(StyleNet) から構成される。事前に学習されたテキスト画像埋め込みモデルである CLIP [16] と本研究で提案する Text-aware Loss を利用して、ネットワーク f のパラメータを最適化し、入力プロンプトによりシーン画像のテキスト領域にスタイル特徴を転送して、最終画像を生成する。

3.1 基本フレームワーク CLIPstyler

本研究のベースモデルとして利用される CLIPstyler を

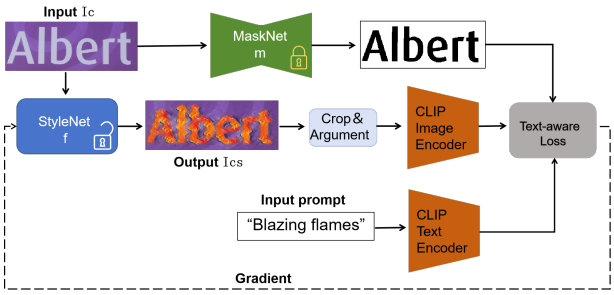

図 3 提案手法 FontCLIPstyler のフレームワーク

紹介する。図 2 のように、CLIPstyler は、入力プロンプトで指定された semantic スタイルを画像に転送することを目的としている。スタイルは自然言語の形式で表現されるため、スタイル画像を必要としない。入力画像 I_c を CNN encoder-decoder モデル f(StyleNet) に入力し、Patch CLIP Loss を用いて f のパラメータを最適化し、スタイル特徴を画像に転送してスタイル画像 I_{cs} を生成する。CLIPstyler で利用された損失関数は式 1 になる。

$$L_{total} = \lambda_d L_{dir} + \lambda_p L_{patch} + \lambda_c L_c + \lambda_{tv} L_{tv} \tag{1}$$

各損失関数について詳しく説明する。Directional CLIP Loss(L_{dir}) は、式 2 のように、入力画像 I_c、入力プロンプト t_{style}、出力画像 $f(I_c)$、コンテンツプロンプト t_{src} をそれぞれ CLIP Encoder でエンコーディングし、方向性を一致させることで、生成された画像が入力プロンプトと同様の semantic スタイルを持つようにする。

$$\begin{aligned} \Delta T &= E_T(t_{style}) - E_T(t_{src}), \\ \Delta I &= E_I(f(I_c)) - E_I(I_c), \\ L_{dir} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|} \end{aligned} \tag{2}$$

Patch CLIP Loss(L_{patch}) は、式 3 のように、生成された画像全体ではなく、ランダムに切り取られた Patch $\hat{I}^i_{cs}$ を使用して Directional CLIP Loss を計算する。さらに、画像の over-stylization を防ぐため、特定の閾値 τ を設定し、この閾値を下回る Patch を無効にする。また、生成された画像に元画像のコンテンツを保持するために、VGG-19 でコンテンツ損失 L_c を計算する。最後に、画像の不規則なピクセルが引き起こす影響を軽減する Total Variation Regularization Loss(L_{tv}) も導入する。

$$\begin{aligned} \Delta T &= E_T(t_{style}) - E_T(t_{src}), \\ \Delta I &= E_I(aug(\hat{I}^i_{cs})) - E_I(I_c), \\ L^i_{patch} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|}, \\ L_{patch} &= \frac{1}{N}\sum_{i}^{n} R(l^i_{patch}, \tau) \\ where\ R(s,\tau) &= \begin{cases} 0, & \text{if } s \le \tau \\ s, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

3.2 FontCLIPstyler

CLIPstyler は、プロンプトの利用により画像のスタイル変換が便利になり、任意のスタイル変更を実現した。しかし、CLIPstyler は画像全体に対してスタイル変換を行い、画像内の特定領域に対するのスタイル変換を行うことはできない。そこで、本研究では CLIPstyler をベースにシーンテキストスタイル変換を実現するための FontCLIPstyler ネットワーク (図 3) を提案する。以下では、本研究で提案するネットワークと損失関数を詳しく説明する。

3.2.1 MaskNet と StyleNet

画像の他の部分に影響を与えずに一部の領域のみにスタイルを変更する場合、簡単かつ効果的な方法は、マスク画像を使用してスタイル変換領域をコントロールすることである。そこで、画像の背景とテキストコンテンツを変更することなく、シーンテキスト画像のテキスト部分のみスタイルを変更することを実現するために、画像内のテキストマスクを抽出するネットワーク MaskNet を提案する。U-Net [18] の計算効率が高く、リアルタイムの画像処理が可能であるため、提案した MaskNet のバックボーンは U-Net を利用する。シーンテキストのスタイル変換を行う際に、MaskNet はフリーズしてシーン画像内にあるテキストのマスク画像を生成する。また、CLIPStyler を参考に、CNN encoder-decoder ネットワーク StyleNet を利用して、本研究で提案する Text-aware Loss で StyleNet のパラメータを最適化することでスタイル付きのシーンテキスト画像を生成する。

3.2.2 Text-aware Loss

Distance Transform Loss [1] は、入力画像の距離変換に基づいて限られた領域内でのスタイル変換を可能にした。したがって、画像内のテキスト領域のみにスタイルを変換するために、本研究では Distance Transform Loss を導入する。Loss は式 4 で定義することができる。具体的には、シーンテキスト画像のテキスト部分の mask 画像を用いて距離変換マップ I_d を作成する。そして、入力画像と出力画像をそれぞれ I_d と乗算し、平均二乗誤差を計算する。

$$L_{distance} = \frac{1}{2}(I_c \cdot I_d - I_{sty} \cdot I_d)^2 \tag{4}$$

入力プロンプトの semantic スタイルを画像に反映するには、Patch CLIP Loss が必要である。しかし、出力画像からランダムに切り取られた Patch にはスタイル変換が不要な背景部分が含まれがちであり、結果として背景にもスタイル特徴が反映されてしまう。この問題を解決するために、本研究では TextPatch CLIP Loss を提案する。具体的には、式 5 のように、まずテキストのマスク画像を用いて入力画像の背景領域画像 $I^i_{b\_patch}$ を切り出し、生成された画像の Patch との余弦類似度を計算する。類似度大きく異なる Patch がテキスト領域に属する Patch と判断し、次に式 6 でテキスト領域に属する Patch のみに対して、Patch CLIP Loss 損失を計算する。

$$\begin{aligned} sim &= 1 - \frac{E_I(\hat{I}^i_{sty}) \cdot (\frac{1}{N}\sum_i^n E_I(I^i_{b\_patch}))}{\left|E_I(\hat{I}^i_{sty})\right| \left|\frac{1}{N}\sum_i^n E_I(I^i_{b\_patch})\right|} \\ where\ \hat{I}^i_{sty} &= \begin{cases} \hat{I}^i_{sty\_b} & if \quad sim < \tau \\ \hat{I}^i_{sty\_t} & otherwise \end{cases} \end{aligned} \tag{5}$$

$$\begin{aligned} \Delta T &= E_T(t_{style}) - E_T(t_{src}), \\ \Delta I &= E_I(aug(\hat{I}^i_{sty\_t})) - E_I(I_c), \\ L^i_{patch} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I|\,|\Delta T|}, \\ L_{patch} &= \frac{1}{N}\sum_i^n l^i_{patch} \end{aligned} \tag{6}$$

CLIPStyler では元画像の前景などのコンテンツを保持するコンテンツ損失 L_c が利用されているため、元画像のテキストスタイルが生成画像にも反映される傾向がある。したがって、元画像のテキストスタイルが生成画像の影響を削減し、背景が変更されないようにするために、本研究では背景再構成損失 L_{recon} を導入する。具体的には、式 7 のように、切り出した Patch の中から背景領域に属する Patch に対して VGG Loss を計算する。

$$\begin{aligned} L^i_{recon} &= \left\| F_{4\_2}(\hat{I}^i_{sty\_b}) - F_{4\_2}(I^i_c) \right\|_2^2 \\ &\quad + \left\| F_{5\_2}(\hat{I}^i_{sty\_b}) - F_{5\_2}(I^i_c) \right\|_2^2 \\ L_{recon} &= \frac{1}{N}\sum_i^n l^i_{recon} \end{aligned} \tag{7}$$

提案した Text-aware Loss の総損失関数は式 8 になる。

$$L_{total} = \lambda_d L_{distance} + \lambda_t L_{patch} + \lambda_b L_{recon} + \lambda_{tv} L_{tv} \tag{8}$$

図 4 提案手法の実世界シーンテキスト画像のスタイル変換結果

4. 実験

4.1 設定

MaskNet は Mostel [15] から集めた 2000 枚の実世界のシーンテキスト画像を使用してトレーニングした。提案モデルを学習する際に、MaskNet をフリーズして、StyleNet のみを学習する。入力のシーンテキスト画像は 512×512 の解像度に変換され、最終に出力結果を元のサイズに戻す。λ_d、λ_t、λ_b、λ_{tv} を 1×10^2、9×10^3、150、2×10^{-3} に設定する。学習率 5×10^{-4}、Adam optimiser を使ってモデルを学習する。Training iteration は 500 に設定し、学習率

は 100 iteration ごとに半減する。モデルの学習は NVIDIA TITAN RTX 1 台を使用し、画像 1 枚あたりの学習時間は約 90 秒から 120 秒であった。

図 5 提案手法を用いた合成シーンテキスト画像のスタイル変換結果

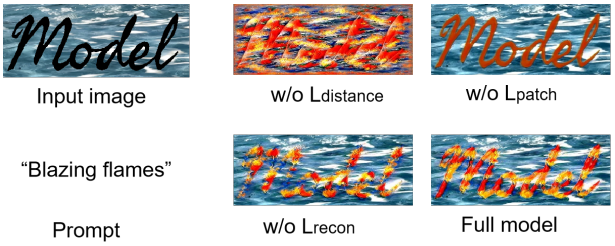

図 6 アブレーション研究の結果

4.2 定性評価

提案手法は異なるプロンプトとシーンテキスト画像を使用して実験を行った。図 4 に示す結果のように、元のテキストコンテンツの可読性を保ちながら、背景を変更せずに、プロンプト指定の semantic スタイルを画像のテキスト部分に反映することに成功している。これらの結果から、提案手法はプロンプトを用いたシーンテキストの任意スタイル変換においての有効性を示している。また、複雑な背景でもテキスト領域のスタイル変換が実現できることを確認するために、高解像度の合成シーンテキスト画像を使用して実験を行った。結果は図 5 のように、背景の細かい質感を保つままで、テキスト領域のスタイル変換を実現できた。

4.3 アブレーション研究

本研究で提案する損失関数 Text-aware Loss の各部分の有効性を検証するために、アブレーション研究を行った。図 6 に示すように、Distance Transform Loss を使用しない場合、スタイルが画像全体にレンダリングしてしまい、テキスト領域への明確なスタイル変換が実現できず、テキストコンテンツも識別不能になる。TextPatch CLIP Loss が欠けている場合、スタイルは画像に反映されない。背景再構成損失を使用しない場合、テキスト領域と背景領域の境界が不明瞭になり、テキストコンテンツの識別が困難になる。また、背景の色などの特徴がテキストに影響を与えてしまう。全ての損失関数を組み合わせたモデルでは、背景とテキストコンテンツを保持したまま、テキスト領域のみにスタイル変換を実現できた。

4.4 他の手法との比較

シーンテキスト編集に関する既存の研究は、テキストコンテンツの変更に取り組んでおり、テキストのスタイルを自由に変更することはできない。そこで、ベース手法 CLIPstyler に加え、本研究の目的に近い、プロンプトを使って画像のスタイル変換を行う手法 Sem-CS [9]、ControlNet [27] との比較を行う。比較結果は図 7 に示している。CLIPstyler と Sem-CS の結果は、スタイル変換が実現されたが、スタイルが画像全体に反映され、テキストのスタイル特徴が目立たない。ControlNet の結果はテキストにスタイルがより良い反映されていたが、背景も大きく変わった。本手法は他の手法と比べ、背景を変更せずにテキスト領域のみのスタイル変換において成功した。

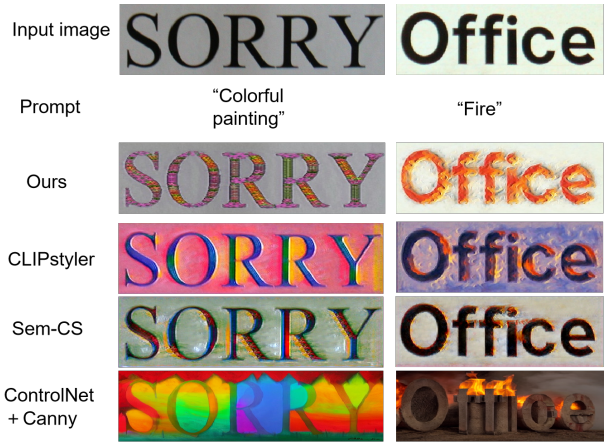

図 7 他の手法との比較

5. 結論

本論文では、シーンテキストのスタイル変換を実現するための FontCLIPstyler を提案する。提案手法は、スタイル参照画像を必要とせず、プロンプトを用いてシーン画像内のテキスト領域のスタイルを自由に変更できた。本研究で提案する Text-aware Loss と MaskNet の利用により、CLIPstyler が画像内の特定の領域へのスタイル変換ができない問題を解決した。実験により、提案手法は画像の背景とテキスト内容を保持しながら、視覚的に魅力的なスタイル付きのシーンテキスト画像を生成することが確認された。本研究はシーンテキストのスタイル変換が実現したが、MaskNet がアルファベットにしか対応していないため、現時点では英語のみに適用し、漢字やカタカナなどの変換はできない。将来的には、他の言語でのシーンテキストスタイル変換の実現に取り組む。

参考文献

[1] Atarsaikhan, G., Iwana, B. K. and Uchida, S.: Contained neural style transfer for decorated logo generation, *2018 13th IAPR International Workshop on Document*

Analysis Systems (DAS), IEEE, pp. 317–322 (2018).

[2] Chen, H., Xu, Z., Gu, Z., Li, Y., Meng, C., Zhu, H., Wang, W. et al.: Diffute: Universal text editing diffusion model, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[3] Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q. and Wei, F.: Textdiffuser: Diffusion models as text painters, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[4] Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L. and Xu, C.: Stytr2: Image style transfer with transformers, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 11326–11336 (2022).

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative adversarial nets, *Advances in neural information processing systems*, Vol. 27 (2014).

[6] Iluz, S., Vinker, Y., Hertz, A., Berio, D., Cohen-Or, D. and Shamir, A.: Word-as-image for semantic typography, *ACM Transactions on Graphics (TOG)*, Vol. 42, No. 4, pp. 1–11 (2023).

[7] Izumi, K. and Yanai, K.: Zero-shot font style transfer with a differentiable renderer, *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, pp. 1–5 (2022).

[8] Ji, J., Zhang, G., Wang, Z., Hou, B., Zhang, Z., Price, B. and Chang, S.: Improving Diffusion Models for Scene Text Editing with Dual Encoders, *arXiv preprint arXiv:2304.05568* (2023).

[9] Kamra, C. G., Mastan, I. D. and Gupta, D.: SEM-CS: Semantic Clipstyler for Text-Based Image Style Transfer, *2023 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 395–399 (2023).

[10] Karras, T., Laine, S. and Aila, T.: A style-based generator architecture for generative adversarial networks, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 4401–4410 (2019).

[11] Krishnan, P., Kovvuri, R., Pang, G., Vassilev, B. and Hassner, T.: Textstylebrush: Transfer of text aesthetics from a single example, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[12] Kwon, G. and Ye, J. C.: Clipstyler: Image style transfer with a single text condition, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 18062–18071 (2022).

[13] Luo, C., Jin, L. and Chen, J.: SimAN: exploring self-supervised representation learning of scene text via similarity-aware normalization, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 1039–1048 (2022).

[14] Ma, J., Zhao, M., Chen, C., Wang, R., Niu, D., Lu, H. and Lin, X.: GlyphDraw: Learning to Draw Chinese Characters in Image Synthesis Models Coherently, *arXiv preprint arXiv:2303.17870* (2023).

[15] Qu, Y., Tan, Q., Xie, H., Xu, J., Wang, Y. and Zhang, Y.: Exploring stroke-level modifications for scene text editing, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, No. 2, pp. 2119–2127 (2023).

[16] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al.: Learning transferable visual models from natural language supervision, *International conference on machine learning*, PMLR, pp. 8748–8763 (2021).

[17] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B.: High-resolution image synthesis with latent diffusion models, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 10684–10695 (2022).

[18] Ronneberger, O., Fischer, P. and Brox, T.: U-net: Convolutional networks for biomedical image segmentation, *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, pp. 234–241 (2015).

[19] Roy, P., Bhattacharya, S., Ghosh, S. and Pal, U.: STEFANN: scene text editor using font adaptive neural network, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 13228–13237 (2020).

[20] Song, Y. and Zhang, Y.: CLIPFont: Text Guided Vector WordArt Generation, *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*, BMVA Press (2022).

[21] Tanveer, M., Wang, Y., Mahdavi-Amiri, A. and Zhang, H.: Ds-fusion: Artistic typography via discriminated and stylized diffusion, *Proc. of IEEE International Conference on Computer Vision*, pp. 374–384 (2023).

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).

[23] Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E. and Bai, X.: Editing text in the wild, *Proceedings of the 27th ACM international conference on multimedia*, pp. 1500–1508 (2019).

[24] Yang, Q., Huang, J. and Lin, W.: Swaptext: Image based texts transfer in scenes, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 14700–14709 (2020).

[25] Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H. and Chen, K.: GlyphControl: Glyph Conditional Control for Visual Text Generation, *Advances in Neural Information Processing Systems*, Vol. 36 (2024).

[26] Yang, Z., Song, H. and Wu, Q.: Generative artisan: A semantic-aware and controllable clipstyler, *arXiv preprint arXiv:2207.11598* (2022).

[27] Zhang, L., Rao, A. and Agrawala, M.: Adding conditional control to text-to-image diffusion models, *Proc. of IEEE International Conference on Computer Vision*, pp. 3836–3847 (2023).