Learning to generate line drawings that convey geometry and semantics

Caroline Chan Frédo Durand Phillip Isola
{cmchan, fredo, phillipi}@mit.edu
MIT

Figure 1. Given a set of photographs, our method is capable of making line drawings in different styles seen above. Our method only requires unpaired data during training.

Abstract

This paper presents an unpaired method for creating line drawings from photographs. Current methods often rely on high quality paired datasets to generate line drawings. However, these datasets often have limitations due to the subjects of the drawings belonging to a specific domain, or in the amount of data collected. Although recent work in unsupervised image-to-image translation has shown much progress, the latest methods still struggle to generate compelling line drawings. We observe that line drawings are encodings of scene information and seek to convey 3D shape and semantic meaning. We build these observations into a set of objectives and train an image translation to map photographs into line drawings. We introduce a geometry loss which predicts depth information from the image features of a line drawing, and a semantic loss which matches the CLIP features of a line drawing with its corresponding photograph. Our approach outperforms state-of-the-art unpaired image translation and line drawing generation methods on creating line drawings from arbitrary photographs. For code and demo visit our webpage carolineec.github.io/informative_drawings

1. Introduction

Through introspection and experimentation, human artists have learned to create line drawings that provide compelling depictions of shape and meaning. A longstanding goal of non-photorealistic rendering is to reproduce this feat and, given an input image, to automatically generate line drawings that are effective at conveying geometry and identity. Manually instilling these qualities into computer-generated line drawings is difficult however because the goals are defined in elusive terms of human perception and cognition. Generating line drawings from photographs presents additional challenges: most photographs lack ground-truth geometry data, and often portray complex scenes with multiple subjects and interactions. Naturally, it would make sense to learn from drawings created by humans or to use humans to evaluate automatic line drawing methods. Unfortunately, the creation of such datasets is challenging and scalability is low.

In this paper, we seek to automatically generate effective line drawings from photographs without requiring paired training data and without requiring human judgment of the implied shape. Our key idea is to view the problem as an encoding through a line drawing and to maximize the quality of this encoding through explicit geometry, semantic, and appearance decoding objectives. Our method approaches line drawing generation as an unsupervised image translation problem which uses various losses to assess the information communicated in a line drawing. This evaluation is performed by deep learning methods which decode

depth, semantics, and appearance from line drawings. The aim is for the extracted depth and semantic information to match the scene geometry and semantics of the input photographs. Appearance preservation follows from cycle consistency [45, 83, 88]. With these objectives, our method is able to create convincing line drawings given unpaired data.

Our main contributions are as follows. We present an unsupervised method for automatic line generation which explicitly instills geometry and semantic information into drawings. We apply our method on many styles of line drawings and present results in Section 4. We also provide analysis of the geometry and semantic information conveyed by our drawings, visual comparisons against several baselines, and an ablation study.

2. Related Work

Line drawings are of particular interest in both art history and psychology. Although studies suggest that the human visual system understands line drawings comparably to photographs [5, 32, 36, 37, 42, 84], it is still unclear why line drawings are effective representations. Several theories exist for this topic, but this area requires further study [29, 30, 69].

There has been extensive work on creating line drawings from 3D geometry. Approaches range from applying image processing to depth and normal maps [8, 68], using geometric features on top of occluding contours [2, 17, 41, 64], to ensembling all geometry-based approaches with deep learning [55]. Although these methods successfully generate line drawings from 3D models, they cannot be applied to arbitrary photographs with unavailable 3D geometry. Furthermore, most methods draw lines in only one style, although Neural Strokes [54] addresses this issue. Instead, our method creates stylized line drawings from 2D photographs which convey 3D geometry.

Most 2D-based line drawing generation methods rely on supervised data. This includes using ground truth stroke or vector graphics data to create drawings [23, 27, 73, 74]. This stroke-based approach is often supported by differentiable architectures which can draw lines [3, 21, 35, 51, 61, 71, 76, 80, 87] and paint [35, 57, 62] with supervision from raster images. Other works focus on conditional line drawing generation given paired images, which are often collected for specific tasks [50, 52, 58, 81]. In contrast, our method handles unpaired data and translates between sketches of different domains.

Our method is most similar to Unpaired Portrait Drawing Generation (UPDG) [81], which creates portrait drawings from unpaired data. UPDG also uses an adversarial image translation setup, but modifies cycle-consistency for drawings, employs a truncation loss, and uses discriminators for the eyes, nose, and mouth. In contrast, our method is built on losses which encourage line drawings to carry meaningful information about geometry and semantics. Our objectives allow us to greatly reduce reliance on cycle consistency (or the appearance reconstruction), and to generate drawings for arbitrary photographs and not just portraits.

Recent work has been successful at text-driven image editing and synthesis with the extensive shared visual-text embedding Contrastive Language-Image Pre-training (CLIP) [16, 65, 66]. CLIPDraw [22] also uses CLIP to create drawings, but with text inputs. This method requires no training, and simply minimizes the CLIP distance between a rasterized set of Bézier curves [51] and the text prompt. CLIPDraw demonstrates that the CLIP embedding can match semantics between text and drawings despite the domain gap. In contrast, previous methods have adapted new architectures to specifically examine semantics in line drawings [4, 85]. Our approach similarly minimizes the distance between inputs and generated drawings in CLIP space, but instead conditions on an input photograph and generates drawings in multiple styles.

Our work also shares similarities to CyCADA [33] in that the output images are trained to semantically match the inputs. However, CyCADA applies this constraint with a pretrained classifier for a translation between source and target data for domain adaptation. In contrast, our semantic constraint makes use of the CLIP embedding, which can richly describe complex scenes.

Given two datasets, modern image translation and style transfer methods can transform images into new domains [24, 31, 38, 40, 88]. Modern approaches can produce high quality results given paired correspondences [11, 20, 38, 77], however large aligned line drawing datasets are scarce. Fortunately, many approaches address image translation for unpaired data, often relying on an adversarial setup [1, 12, 43, 45, 63, 70, 78, 78, 83, 86, 88]. Other methods translate images between domains by separating style and content [34, 39, 56]. Although these approaches are very successful at artistic style transfer and translating between rich domains with shape changes (e.g. dogs to cats, anime to selfies), they still generate create sparse line drawings which are missing key strokes.

3. Method

Our goal is to train a model to automatically generate line drawings of arbitrary photographs given a dataset of photographs and an unpaired dataset of line drawings. We formulate this problem as unpaired image translation between domain A which contains photographs, and domain B which represents line drawings of a particular style. Most previous approaches solely consider preserving photographic appearance in the line drawing through cycle consistency. Instead, our method further directs this translation through objectives which assess the geometry and semantic information communicated by line drawings. This setup is

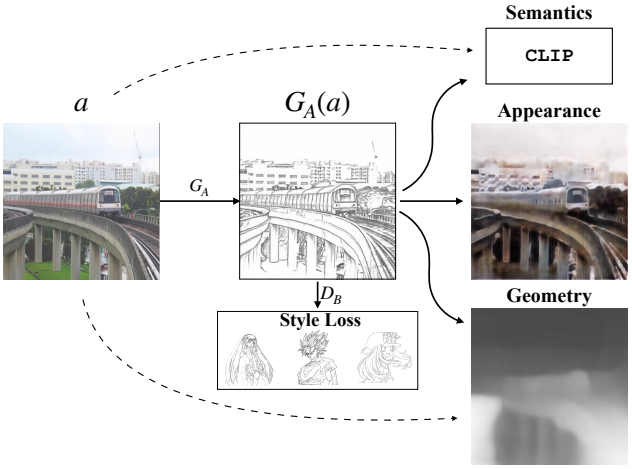

Figure 2. Given a photograph a, our model trains network G_A to synthesize line drawing $G_A(a)$ via four main losses. Adversarial style loss with discriminator D_B encourages generated line drawings to match the style of the training set. The CLIP, appearance, and geometry losses enforce that the line drawing communicates effective semantic, appearance, and geometry respectively.

shown in Figure 2. We show in Section 4 that these new losses are essential for creating meaningful drawings.

We use an adversarial training setup with generator networks G_A, G_B and discriminators D_A, D_B for domains A and B respectively. The geometry objective is implemented through a pretrained depth network which predicts depth maps from line drawings, and imposes a supervised loss on the depth outputs. This loss encourages our model to draw lines in geometrically important locations (e.g. occluding contours). Secondly, we introduce a CLIP [66] loss to add semantic meaning into the generated line drawings. Because arbitrary photographs often display complex scenes, we use the visual CLIP embedding which captures semantic details quite well. We then impose that the CLIP embedding of the line drawing is similar to the CLIP embedding of the original photograph. We also use a weakly weighted cycle consistency loss to preserve appearance information.

3.1. Losses

The adversarial loss encourages generated images to belong to their respective domains [25]. The loss for each domain using the LSGAN setup [59] is formulated below.

$$\begin{aligned} L_{GAN} = \quad & \mathbb{E}_{a\sim A}[D_A(a)^2] + \mathbb{E}_{b\sim B}[\big(1 - D_A(G_B(b))\big)^2] \\ + \quad & \mathbb{E}_{b\sim B}[D_B(b)^2] + \mathbb{E}_{a\sim A}[\big(1 - D_B(G_A(a))\big)^2] \end{aligned} \tag{1}$$

The geometry objective maximizes depth information in generated line drawings during training. We observe that line drawings are often effective conveyors of 3D shape, and apply this property during training. Given a substantial dataset of line drawings, a model may learn this trait without any explicit supervision. However, current methods without such geometric constraints fail to place lines in meaningful places (see Section 4). Domain gaps between the dataset of photographs and line drawings are also obstacles. Instead, we propose a geometric constraint which supervises depth predictions from line drawings.

To supervise depth predictions from line drawings, it is necessary to obtain depth maps for the photographic inputs. Unfortunately, ground truth depth information is usually unavailable for most datasets. However, recent methods are very successful at producing high resolution depth maps for photographs. This advance allows us to use pseudo-ground truth depth maps obtained from a state of the art depth prediction network F; in practice we use the network from [60], which is based on MiDaS [67]. We note that pseudo-ground truth maps for photographs are only required for training, and not at test time.

A simple way to supervise geometry predictions would be to introduce network G_{Geom} to predict depth maps from line drawings during training. However, this approach has several issues. Training G_{Geom} to learn depth from synthetic line drawings may encourage line drawing generator G_A to instill depth information in an unwanted form, such as an imperceptible signal [14]. We want to avoid accidentally embedding invisible information into our line drawings. Using pretrained depth network F on line drawings is not an option because of the domain gap.

We propose instead to learn to infer depth from image features which are commonly shared between photographs and line drawings. Specifically, we pretrain a network G_{Geom} to predict depth given ImageNet [18] features. Such features, especially in early layers, are useful for transfer learning [47]. This scenario hopes to avoid the invisible signal issue by first encoding line drawings into a shared representation with photographs, and then applying a network which has learned depth from photographic features.

To obtain image features, we input photographs into pretrained Inception v3 [75] network and extract features from the Mixed 6b node (see supplemental). We denote the extracted features at this layer for input a as $I(a)$. After pretraining, network G_{Geom} provides depth map predictions for line drawings. In practice, we finetune G_{Geom} while training line drawing generation.

The geometry loss is formulated below. Given photograph a, we first input a into state of the art depth network F and obtain pseudo-ground truth depth map $F(a)$. We then generate line drawing $G_A(a)$ and extract its ImageNet features $I(G_A(a))$. These features are then passed to pretrained depth network G_{Geom} to produce depth map prediction $G_{Geom}(I(G_A(a)))$. This depth prediction is then compared to the pseudo-ground truth depth map $F(a)$. Further details and depth reconstructions are in the supplementary.

$$L_{geom} = \|G_{Geom}(I(G_A(a))) - F(a)\| \tag{2}$$

The semantics loss is implemented by minimizing the distance between the CLIP embeddings of the input photograph and the generated line drawing. The goal of this objective is to convey semantic information from the original photograph into its corresponding synthesized line drawing. In computer vision, semantics are often learned in the form of labels and segmentation maps. However, these representations are limited in capacity to specific domains or objects. To encode semantic information from entire scenes, we use the shared visual-text embedding CLIP [66], which captures rich semantic information in both photographs and art [16, 22]. We then penalize the distance in CLIP space between the generated line drawing and the original photograph. The objective is formulated below.

$$L_{\texttt{CLIP}} = \|\texttt{CLIP}(G_A(a)) - \texttt{CLIP}(a)\| \quad (3)$$

The appearance loss (or cycle consistency) has been used to encode input appearance through image translation [45, 88]. The appearance loss for each direction of the mapping is below.

$$L_{cycle} = \|G_B(G_A(a)) - a\| + \|G_A(G_B(b)) - b\| \quad (4)$$

3.2. Full Objective

Our full objective is:

$$\begin{aligned} L = \quad & \lambda_{\texttt{CLIP}} L_{\texttt{CLIP}} + \lambda_{geom} L_{geom} \\ & + \lambda_{GAN} L_{GAN} + \lambda_{cycle} L_{cycle} \end{aligned} \quad (5)$$

In practice we set $\lambda_{\texttt{CLIP}} = 10$, $\lambda_{geom} = 10$, $\lambda_{GAN} = 1$, $\lambda_{cycle} = 0.1$.

Implementation We use an encoder-decoder generator architecture with Res-Net blocks in the middle [28, 40, 88], and a patch-based discriminator [38]. The architecture for pretrained depth network G_{Geom} is based on the Global Generator from pix2pixHD [77] and further detailed in the supplemental material. We use MSE error for the CLIP loss and $L1$ distance for the appearance and geometry losses. We use Adam [46] to optimize with a learning rate of 0.0002 and train for at least 30 epochs with batch size 6.

4. Experiments

We evaluate our described approach and provide qualitative and quantitative comparisons for both general photographs and portraits in multiple styles.

4.1. Line Drawings from Photographs

Our first evaluation task is to generate line drawings from photographs of arbitrary scenes. Below we describe the datasets for training and evaluation.

Datasets For training, our method requires a dataset of photographs and a separate dataset of line drawings. We train on a randomly selected $10,000$ image subset of the Common Objects in Context (COCO) [53] dataset which contains a variety of scenes. For evaluation, we create line drawings from photographs in the MIT-Adobe FiveK dataset [7]. This dataset contains high quality images of many subjects (landscapes, buildings, people, etc).

We train multiple models with different styles of line drawings. Examples for each style are shown in Figure 3. Quantitative evaluations are performed for two styles of line drawings: **1) The Contour Drawings dataset [50]** contains $5,000$ drawings for various scenes (often with humans or dogs). **2) The Anime Colorization dataset [44]** consists of $14,224$ sketches of various anime characters. Qualitative results in the style of OpenSketch [26] and artist drawings from Cole et al. [15] are shown in Figure 3.

Comparison methods We compare our approach to state-of-the-art unpaired image-to-image translation methods for the photograph to line drawing task. These methods include: **1) CycleGAN [88]** uses an appearance loss and a patch-based discriminator [38]. **2) TSIT [39]** creates images by combining features from separate content and style streams. **3) U-GAT-IT [43]** uses an attention module and auxiliary classifier and cycle consistency. **4) ACL-GAN [86]** relaxes strict pixel cycle consistency into distributional level consistency **5) Unpaired Portrait Drawing Generation (UPDG) [82]** creates line drawings in multiple styles for portrait drawings. This method builds upon CycleGAN with discriminators for facial features, a truncation loss, and a modified cycle loss using HED images [79]. For the photograph task, we do not include the face discriminators as they do not apply to arbitrary photographs without human subjects. We also provide qualitative comparisons with SPatchGAN [70] and Council-GAN [63] in Figure 4.

Qualitative comparison Figure 4 shows compares our method to previous work in two styles. Other methods commonly fail to place lines in meaningful locations, whereas our drawings have recognizable features and boundaries. Some methods such as SPatchGAN, Council-GAN, and ACL-GAN attempt to strictly stay close to the training set domain. This is most noticeable for the Anime style, as these approaches often produce drawings which resemble anime characters over the input photographs.

User Study We conduct a user study to perceptually compare our approach with other methods. In this study, participants were shown a reference photograph, and two line drawings of the same photograph made by different methods. Users were then asked to select the line drawing that best depicts the input photograph. For this study we showed users up to 100 images and there were 184 unique participants. 1000 judgments were made for each comparison. Ta-

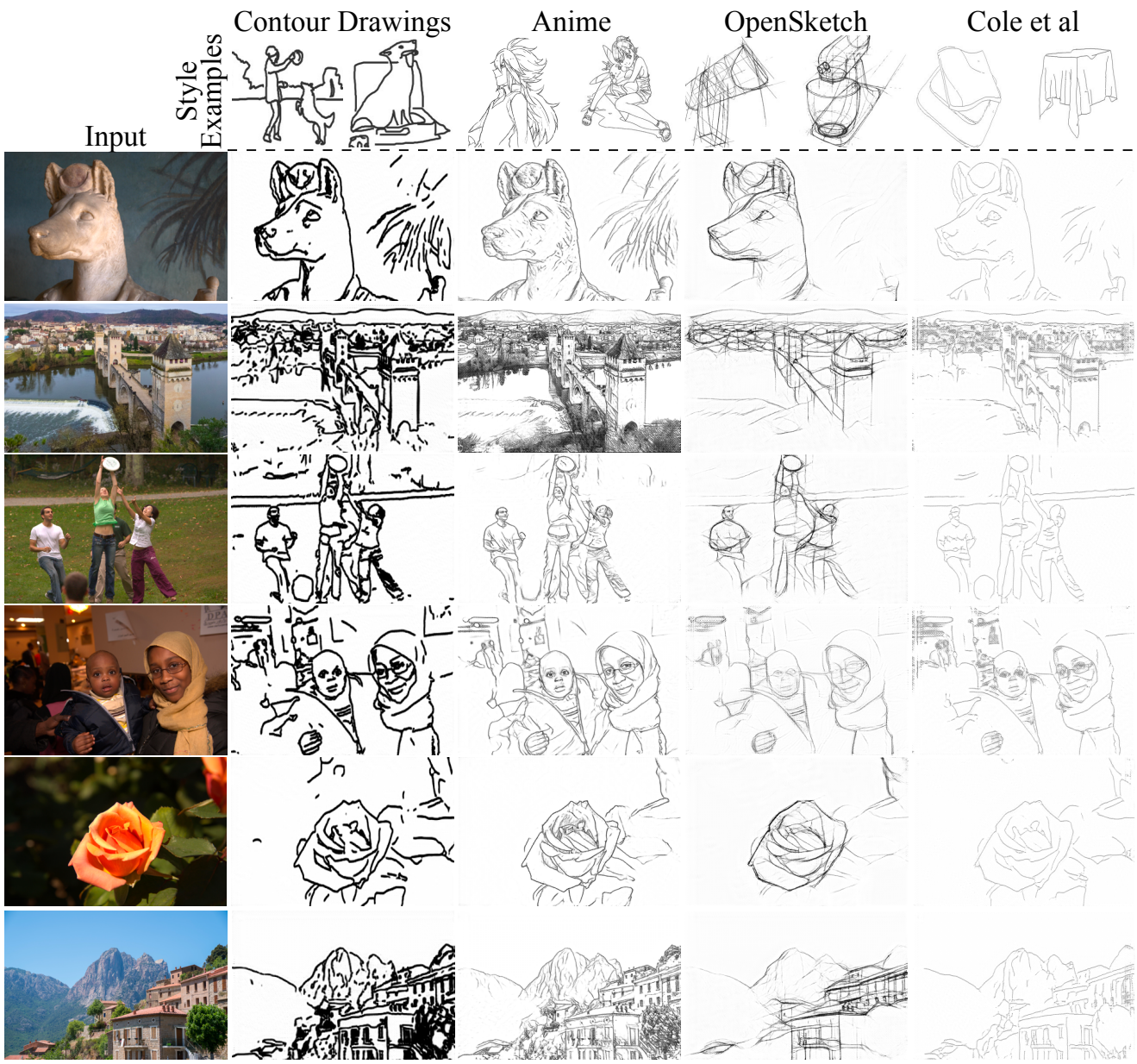

Figure 3. Results of our method in four different styles.

	Contour Drawings	Anime	Total
CycleGAN [88]	98.7%	87.3%	93 %
TSIT [39]	99.6%	95.3%	97.5%
U-GAT-IT [43]	99.5%	97.3%	98.4%
ACL-GAN [86]	100%	97.5%	98.8%
UPDG [82]	98.9%	96.7%	97.8%

Table 1. User study results comparing to different unpaired translation methods. We report the percentage of times users preferred our approach over the other methods.

ble 1 reports the percentage users chose line drawings from our method over various baselines. Users overwhelmingly preferred line drawings created by our method in all cases.

Ablation Study We perform an ablation study to verify the inclusion of each loss. Three versions of our model are trained: without the geometry loss, without the CLIP loss, and without the appearance or cycle loss. We compare each ablation to our full method. We use the perceptual study setup described above and report the percentages users selected our full method over each ablation in Table 2. The CLIP loss was essential for all styles, while the Contour Drawings style relies on the depth loss much more than the Anime style. The appearance loss improves results slightly.

Figure 5 shows qualitative examples from all ablations. The CLIP loss adds the most lines. In some cases, styles

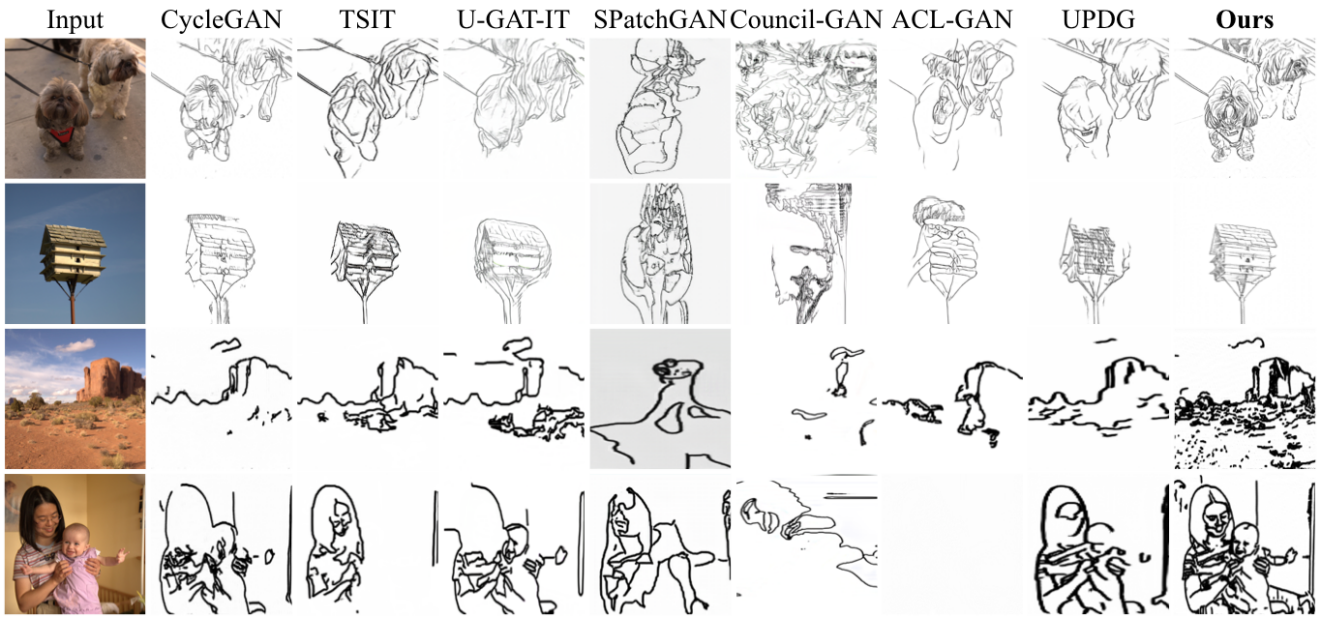

Figure 4. Comparison with other methods. *Left to right:* Input photograph, CycleGAN, TSIT, U-GAT-IT, SPatchGAN, Council-GAN, ACL-GAN, UPDG, and Our approach. All methods are trained using the same data on two styles of line drawings. Our method produces the most detailed drawings capturing important aspects of the original photograph.

	Contour Drawings	Anime	Total
Without depth	92.2%	48.3%	70.3%
Without CLIP	98.9%	84.9%	92%
Without Cycle Consistency	87.0%	64.9%	76%

Table 2. User study results for the ablation study. We report the percentage users chose the full method over the ablations.

	Contour Drawings	Anime	Total
CycleGAN	58.0%	65.1%	62.0%
Ours	68.4%	66.8%	67.6%
Photograph	–	–	70.3%

Table 3. User study results for relative depth prediction. We report the percentage of times users chose the closer point correctly for each baseline. For both styles, users correctly inferred relative depth more often in drawings from our method over CycleGAN.

with a high density of lines may totally rely on the CLIP loss. We find this situation to be the case for the Anime style, whose 'without depth' ablation is comparable to the full method. The depth loss is most useful for sparse styles such as the Contour Drawings style, where it adds occluding contours and textures. We note that the semantic loss improves geometry, and depth information can help semantics as well. The cycle loss improves result quality by preserving appearance aspects such as textures and outlines. However, removing the cycle loss does not qualitatively affect results significantly.

Evaluating Geometry and Semantics in Drawings We design two experiments to evaluate the depth and semantic information conveyed in the generated line drawings. To examine depth information, we conduct a user study to assess if humans can correctly infer relative depth from our drawings. Participants viewed an image with two randomly placed points and were asked to identify the point closest to the camera, similarly to [13]. We perform this evaluation on drawings from our method, CycleGAN, and on photographs. Table 3 reports the percentage each baseline agreed with the pseudo-ground truth depth predictions. In general, users inferred the correct relative depth more often in our drawings, especially for the Contour Drawings style. For the Anime style, relative depth predictions were better for our results by a slim margin. This result complements the ablation study, where the depth loss was not as effective for the Anime style. If relative depth can already be inferred from CycleGAN (despite lower drawing quality), then the geometry objective may not have much impact. In contrast, the depth loss greatly improves both relative depth predictions and drawing quality for the Contour Drawing style.

	Contour Drawings	Anime	Total	Unrecognizable
CycleGAN	0.7436	0.8074	0.7799	26.7%
Ours	0.8160	0.8371	0.8274	13.7%
Photograph	–	–	0.8804	0.02%

Table 4. Mean cosine similarity between captions describing line drawings and captions describing the input photographs. The last column reports the percentage of images that users could not identify. Our line drawings are more easily described and recognizable.

To assess semantic meaning, we show users a photograph and ask them to write a one sentence caption for the

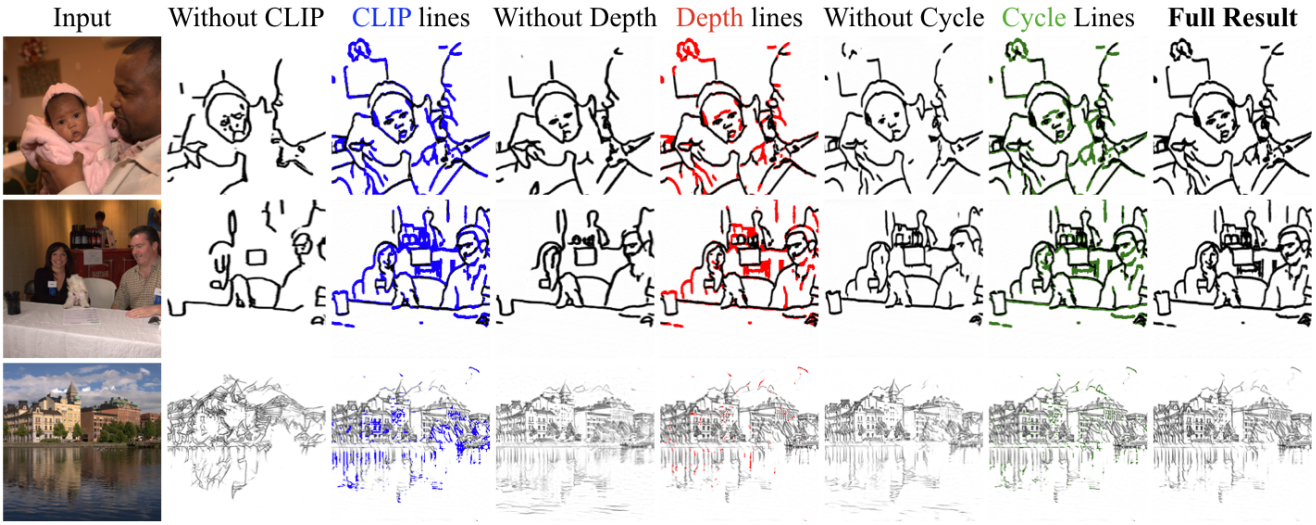

Figure 5. Ablations of our method, and our full result. For each ablations, we show the lines added to get the full result by including each loss. These lines are in blue for CLIP, red for depth, and green for appearance. The CLIP loss adds the most lines, while the depth loss adds more information and occluding contours in the second row. The appearance loss adds small strokes and shading for the Anime style.

image. Participants were also given the option to designate images as unrecognizable. Users viewed results from our method, CycleGAN, and photographs. Each caption is encoded in CLIP space and then compared to the mean CLIP embedded photograph caption using cosine similarity. Table 4 reports the mean cosine similarities and the percentage of unrecognizable images. In all cases our method produces more accurate descriptions and recognizable drawings.

4.2. Line Drawings from Portraits

While our method was not designed specifically for portraits, we compare to methods specialized for this task. We use two main settings for comparison. Firstly, we compare to other methods directly on styles they present. Then we provide a second comparison where we train our model on unpaired portraits from the Helen Facial Feature Dataset [48] in the style of the APDrawings dataset [81]. Details for each dataset are provided in the supplemental.

Comparisons **1) APDrawingGAN [81]** uses supervised adversarial training to create line drawings in the style of the paired APDrawings. In one comparison, we train our model on APDrawings directly. This setting disadvantages our method because we do not use paired supervision. However, our method can use unpaired data and we exploit this property in the next case. We then use portraits from the Helen dataset to train a separate model, while keeping the drawing style of APDrawings. Our second comparison evaluates our method trained on the Helen dataset against supervised APDrawingGAN results.

2) Unpaired Portrait Drawing Generation (UPDG) [82] is described in Section 4.1. In the first setting, we compare to a pretrained UPDG model in the style of illustrators Charles Burns [6] and Yann Legendre [49] (style 1 from [82]). We train our model from scratch on an approximation of these datasets (see supplemental), and evaluate on the Helen test set. Secondly, we train both our approach and UPDG from scratch to create portraits from the Helen dataset in the style of APDrawings. We then compare on test portraits from APDrawings.

Qualitative comparison Figure 6 shows portrait drawings created with APDrawings from all methods. APDrawingGAN produces reasonable results, while UPDG struggles with the line art style. We achieve decent results training on APDrawings, but quality drastically improves by training on the Helen dataset. Both our method and UPDG create high quality drawings in style 1 (see supplemental).

User Study We perform a user study for all portrait comparisons. Participants were shown a portrait and two line drawings from different methods and asked to select the drawing which best depicts the subject in the portrait. Table 5 reports the percentage of times users chose our approach over the baselines. In case 1, users preferred the supervised APDrawingGAN over our method (trained on APDrawings), but found our method (trained on Helen) preferable or comparable in case 2. In general, UPDG struggles with the APDrawings style, and overall users slightly preferred our method for style 1.

5. Discussion

Loss Formulations We explored several variants of the geometry and semantic losses in initial experiments. This includes using normal maps and multi-view consistency. We found the normal maps helpful for 3D shapes, however

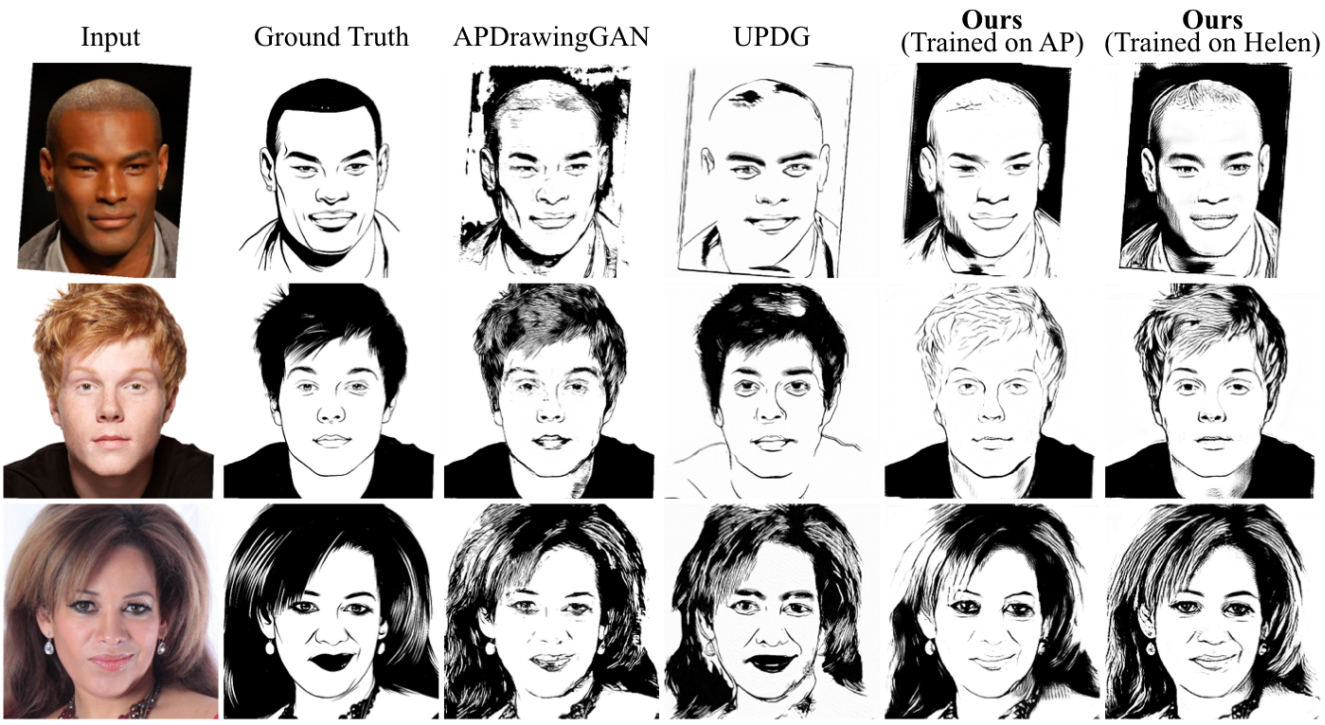

Figure 6. Results for several methods on APDrawings test data. *Left to right:* Portrait photograph, artist's drawing, APDrawingGAN, UPDG (trained on Helen), Our result (trained on APDrawings), our result (trained on Helen). All methods were trained with the APDrawings line art style. Our approach produces accurate and well formed drawings.

	Case 1	Case 2
APDrawingGAN [81]	36.7%	60.1%
UPDG [82]	64.2%	94.8%

Table 5. Perceptual study results for portrait comparisons. We report the percentage users chose our approach over each baseline. Case 1 compares both baselines on their datasets and styles. In case 2, we train our model on Helen in the style of APDrawings and compare to baselines trained on the same style.

normal estimates are often noisy for photographs. Novel view prediction and using other 3D approaches are directions we hope to explore in future work. We selected depth prediction [60] due to its robustness on photographs, and because we can reliably obtain depth predictions from image features that also can be extracted from line drawings. For the semantic loss, we explored finetuning image classifiers and segmentation networks on drawings and comparing intermediate features from these networks [10, 18]. For a visual comparison, see the supplemental material.

Limitations Our method is built on some limiting assumptions. We rely on pseudo-ground truth depth maps from a pretrained network for geometry supervision. Because we essentially distill this pretrained depth prediction network, our model has similar failure cases and biases.

Our model produces meaningful line drawings for many styles, but has failure cases shown in the supplemental. Our method is based on the hypothesis that a good line drawing accurately conveys depth and semantics, however some styles focus on the essence of the scene and not precision. We also struggle with certain lighting conditions and textures. Overall, the CLIP loss drives results to look more 'photographic,' which may or may not be desirable. In some cases, this causes results to converge to grayscale photos.

Negative Impacts As with most data-driven techniques, our approach can learn bias in training. For instance, the Anime sketch dataset in Section 4 contains drawings of mostly feminine subjects. In addition, artistic datasets (such as the full Anime dataset used for creating line drawings) may contain sensitive content (e.g. nudity, weapons) whose influence could be visible in the output.

Conclusion Our approach creates compelling line drawings given unpaired data. This paper views line drawings as encodings of geometry, semantics, and appearance from real scenes. We built these ideas into a method which explicitly evaluates these properties through depth prediction, CLIP features, and image reconstruction to create line drawings from photographs.

Acknowledgements We would like to thank Hyojin Bahng for proofreading the paper. This work was partially supported by a Packard Fellowship to PI, and the National Science Foundation under Grant No. 2105819.

References

[1] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018. 2

[2] Pierre Bénard and Aaron Hertzmann. Line drawings from 3d models: a tutorial. *Foundations and Trends in Computer Graphics and Vision*, 11(1-2):159, 2019. 2

[3] Mikhail Bessmeltsev and Justin Solomon. Vectorization of line drawings via polyvector fields. *ACM Transactions on Graphics (TOG)*, 38(1):1–12, 2019. 2

[4] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Ghttps://www.kaggle.com/ktaebum/anime-sketch-colorization pairryaditskaya, and Yi-Zhe Song. Pixelor: A competitive sketching ai agent. so you think you can sketch? *ACM Trans. Graph.*, 39(6), 2020. 2

[5] Irving Biederman and Ginny Ju. Surface versus edge-based determinants of visual recognition. *Cognitive psychology*, 20(1):38–64, 1988. 2

[6] Charles Burns. Cover portraits for the believer, 2003-2013. Adam Baumgold Gallery, 2013. 7, 17

[7] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011. 4

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 2

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 14

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 8, 13

[11] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 2

[12] Runfa Chen, Wenbing Huang, Binghui Huang, Fuchun Sun, and Bin Fang. Reusing discriminators for encoding: Towards unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8168–8177, 2020. 2

[13] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29:730–738, 2016. 6

[14] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 3

[15] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. Where do people draw lines? In *ACM SIGGRAPH 2008 papers*, pages 1–11. 2008. 4, 16

[16] Katherine Crowson. Vqgan-clip. https://github.com/nerdyrodent/VQGAN-CLIP, 2021. 2, 4

[17] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers*, pages 848–855. 2003. 2

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 8, 14

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 14

[20] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 2

[21] Kevin Frans and Chin-Yi Cheng. Unsupervised image to sequence translation with canvas-drawer networks. *arXiv preprint arXiv:1809.08340*, 2018. 2

[22] Kevin Frans, LB Soros, and Olaf Witkowski. Clipdraw: exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*, 2021. 2, 4

[23] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1666–1675, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[24] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014. 3

[26] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia Pont, Fredo Durand, and Adrien Bousseau. Opensketch: A richly-annotated dataset of product design sketches. *ACM Transactions on Graphics (TOG)*, 38(6):232, 2019. 4

[27] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 4

[29] Aaron Hertzmann. Why do line drawings work? a realism hypothesis. *Perception*, 49(4):439–451, 2020. 2

[30] Aaron Hertzmann. The role of edges in line drawing perception. *Perception*, 50(3):266–275, 2021. 2

[31] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001. 2

[32] Julian Hochberg and Virginia Brooks. Pictorial recognition as an unlearned ability: A study of one child's performance. *The American Journal of Psychology*, 75(4):624–628, 1962. 2

[33] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 2

[34] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[35] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8709–8718, 2019. 2

[36] DH Hubel and TN Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574, 1959. 2

[37] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968. 2

[38] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2, 4, 13

[39] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. TSIT: A simple and versatile framework for image-to-image translation. In *ECCV*, 2020. 2, 4, 5

[40] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2, 4, 13

[41] Tilke Judd, Frédo Durand, and Edward Adelson. Apparent ridges for line drawing. *ACM transactions on graphics (TOG)*, 26(3):19–es, 2007. 2

[42] John M Kennedy and Abraham S Ross. Outline picture perception by the songe of papua. *Perception*, 4(4):391–406, 1975. 2

[43] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *International Conference on Learning Representations*, 2020. 2, 4, 5, 13

[44] Taebum Kim. Anime sketch colorization pair. https://github.com/vijishmadhavan/ArtLine, 2018. 4

[45] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning*, pages 1857–1865. PMLR, 2017. 2, 4

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[47] Simon Kornblith, Honglak Lee, Ting Chen, and Mohammad Norouzi. What's in a loss function for image classification? *arXiv preprint arXiv:2010.16402*, 2020. 3, 14

[48] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *European conference on computer vision*, pages 679–692. Springer, 2012. 7, 17

[49] Yann Legendre. Portraits. http://www.yannlegendre.com/project/portraits/, 2022. 7

[50] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1403–1412. IEEE, 2019. 2, 4

[51] Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15, 2020. 2

[52] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, and Ming-Hsuan Yang. Im2pencil: Controllable pencil illustration from photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[53] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 14

[54] Difan Liu, Matthew Fisher, Aaron Hertzmann, and Evangelos Kalogerakis. Neural strokes: Stylized line drawing of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[55] Difan Liu, Mohamed Nabail, Aaron Hertzmann, and Evangelos Kalogerakis. Neural contours: Learning to draw lines from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5428–5436, 2020. 2

[56] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsueprvised image-to-image translation. In *arxiv*, 2019. 2

[57] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6598–6607, 2021. 2

[58] Vijish Madhavan. Artline. https://www.kaggle.com/ktaebum/anime-sketch-colorization-pair, 2020. 2

[59] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 3

[60] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. 2021. 3, 8, 13, 14

[61] Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, Ruomei Wang, Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, et al. General virtual sketching framework for vector line art. *ACM Trans. Graph*, 1(1), 2021. 2

[62] Reiichiro Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv preprint arXiv:1904.08410*, 2019. 2

[63] Ori Nizan and Ayellet Tal. Breaking the cycle - colleagues are all you need. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2020. 2, 4

[64] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. In *ACM SIGGRAPH 2004 Papers*, pages 609–612. 2004. 2

[65] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 2021. 2, 3, 4, 13, 14

[67] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv:1907.01341*, 2019. 3, 14

[68] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 197–206, 1990. 2, 14

[69] Bilge Sayim and Patrick Cavanagh. What line drawings reveal about the visual brain. *Frontiers in human neuroscience*, 5:118, 2011. 2

[70] Xuning Shao and Weidong Zhang. Spatchgan: A statistical feature based discriminator for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6546–6555, October 2021. 2, 4

[71] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG)*, 37(1):1–13, 2018. 2

[72] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 13

[73] Dmitriy Smirnov, Matthew Fisher, Vladimir G. Kim, Richard Zhang, and Justin Solomon. Deep parametric shape predictions using distance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[74] Yi-Zhe Song. Bézier sketch: A generative model for scalable vector sketches. *Computer Vision–ECCV 2020*, 2020. 2

[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3, 13, 14

[76] Alexander Wang, Mengye Ren, and Richard Zemel. Sketchembednet: Learning novel concepts by imitating drawings. *arXiv preprint arXiv:2009.04806*, 2020. 2

[77] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2, 4, 14

[78] Shaoan Xie, Mingming Gong, Yanwu Xu, and Kun Zhang. Unaligned image-to-image translation by learning to reweight. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14174–14184, 2021. 2

[79] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 4

[80] Xuemiao Xu, Minshan Xie, Peiqi Miao, Wei Qu, Wenpeng Xiao, Huaidong Zhang, Xueting Liu, and Tien-Tsin Wong. Perceptual-aware sketch simplification based on integrated vgg layers. *IEEE transactions on visualization and computer graphics*, 27(1):178–189, 2019. 2

[81] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2019. 2, 7, 8, 17

[82] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Unpaired portrait drawing generation via asymmetric cycle mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '20)*, pages 8214–8222, 2020. 4, 5, 7, 8, 13, 14

[83] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017. 2

[84] Albert Yonas and Martha E Arterberry. Infants perceive spatial structure specified by line junctions. *Perception*, 23(12):1427–1435, 1994. 2

[85] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural

network that beats humans. *International journal of computer vision*, 122(3):411–425, 2017. 2

[86] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. In *European Conference on Computer Vision*, pages 800–815. Springer, 2020. 2, 4, 5

[87] Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2018. 2

[88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 4, 5, 13

6. Supplemental Material

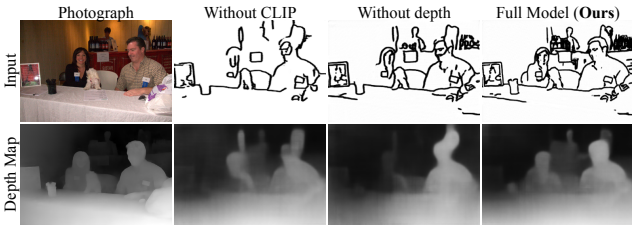

Figure 7. Depth predictions for various inputs. From left to right, we show the input photograph, the ablation trained without the semantic loss, the ablation trained without the depth loss, and our full model. The bottom row shows depth maps predicted from the images on the top row.

6.1. More Results

In this section we provide more results. We show further results on four styles of line drawings in Figure 10. The OpenSketch and Cole et al tend to focus on the center of the image and sometimes are missing lines.

6.1.1 Depth Reconstruction

We show depth predictions from our line drawings in Figure 11 and compare to the pseudo-ground truth. Our depth prediction maps are coarse, but often capture key elements and relative depth of a scene. However, they are not always accurate as seen in the first and second rows. Our depth network sometimes interprets boundary lines as close objects rather than part of the background (such as the doorway in the first row of Figure 11). Additionally, we find depth information is not so useful for scenes where all objects are either very close or very far from the camera. Adding the semantic loss also increases predicted depth map quality.

We provide a comparison of the depth maps predicted from line drawings in Table 6 and report the mean squared error for different ablations. Depth maps from state of the art pretrained model [60] are used as a pseudo ground truth. We find that the reported errors are consistent with our results - adding the depth loss improves depth maps for the contour drawing style, whereas the anime style already portrays depth information well without the depth loss. Figure 7 shows qualitative results and pseudo ground truth depth maps.

	Contour Drawings	Anime	Total
Without depth	0.0530	0.0400	0.0465
Full model (with depth)	0.0506	0.0418	0.0462

Table 6. Predicted depth map MSE errors for different ablations of our model.

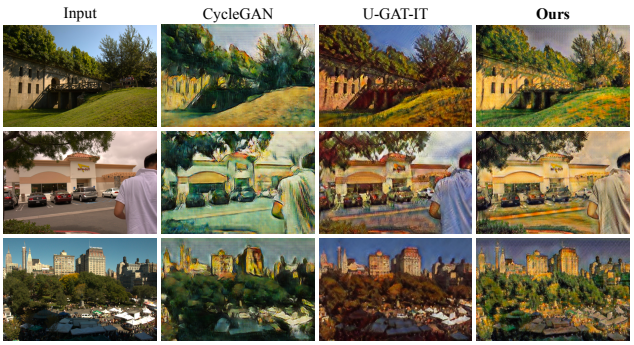

Figure 8. Our method designed for line drawing generation using Cezanne paintings as the style produces results comparable with other methods. The geometry and semantics loss works best with sparser styles. *Left to right:* input photograph, CycleGAN results, U-GAT-IT results, and our method. Results are overall comparable.

6.1.2 Painterly styles

Although our method is specifically designed for line drawings where the geometry and semantics increase the quality of sparser styles, we try creating images in the style of Cezanne paintings. Results are seen in Figure 8 where we provide comparisons to CycleGAN [88] and U-GAT-IT [43]. All approaches handle the Cezanne style well and produce comparable results.

6.1.3 Portraits in Style 1

We provide further comparisons for methods which specifically create portrait drawings, even though our method is designed for arbitrary photographs. Figure 9 provides visual examples for the comparison to UPDG in style 1 from their paper [82]. Overall, both methods create nice portraits.

6.2. Other loss variants

Figure 12 compares semantic features between generated line drawings and input photographs from DeepLabv3 [10], Inception v3 [75], and CLIP [66]. Using DeepLabv3 features created confusing drawings. Inception v3 and CLIP features both create reasonable drawings, however we chose the CLIP embedding mainly due to its better handling of lighting, textures, backgrounds, and faces.

6.3. Network Architectures

Generators and Discriminators We use the encoder-decoder generator with 3 Res-Net blocks [72] from [40, 88] for networks G_A and G_B. For the disciminators, we use a PatchGAN architecture [38] with a receptive field of 70×70. The code for these networks are available for academic use under their licenses.

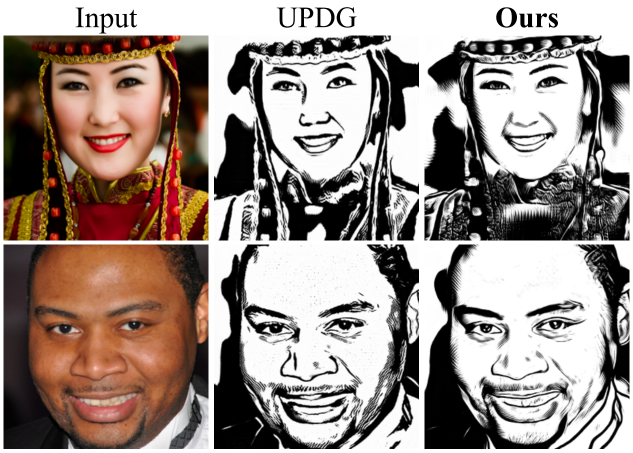

Figure 9. Comparison between UPDG results on style 1 from [82]. Our method can be applied to arbitrary photographs whereas UPDG specifically creates portraits. *Left to right:* Portrait photograph, UPDG, and our method. Although we were unable to exactly match the training data, both methods were trained using portraits 'in the wild' and using line art style 1 from [82]. Both approaches often produce nice drawings of the subject.

Layer Type	Padding	Kernel Size	Stride	Normalization	Activation	Input, Output Channels
Conv2D	4	7×7	1	BatchNorm	ReLU	768,512
ConvTranspose2D	0	4×4	2	BatchNorm	ReLU	512,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ResNet Block	1	3×3	1	BatchNorm	ReLU	256,256
ConvTranspose2D	1	3×3	2	BatchNorm	ReLU	256,128
ConvTranspose2D	1	3×3	2	BatchNorm	ReLU	128,64
ConvTranspose2D	1	3×3	2	BatchNorm	ReLU	64,64
Conv2D	3	7×7	1	BatchNorm	Tanh	64,3

Table 7. Architectures for G_{Geom} which translates ImageNet features into depth maps.

Depth Networks To obtain pseudo-ground truth depth maps, we use the output of a pretrained depth prediction system F presented by Miangoleh et al [60] which is built around MiDaS [67]. These models are available under academic and MIT licenses respectively.

The network architecture for image features to depth map network G_{Geom} is based off of the Global Generator presented by Wang et al in pix2pixHD [77]. Namely, the beginning layers of the network have been modified to account for the input image features. The individual layers are detailed in Table 7. G_{Geom} is first pretrained on image features from real photographs to produce depth maps as seen in Figure 13.

To obtain image features, we use an Inception v3 [75] network which has been pretrained on ImageNet [18]. Specifically, we extract features from the Mixed 6b node. We chose this layer for a several reasons. Firstly, previous work has indicated that earlier network features are more important for transfer learning [47], and we find this situation applicable to line drawings. Secondly, we conducted a nearest neighbor classification experiment with the ShapeNet dataset [9] (terms of use cover research purposes). This dataset consists of many 3D models of labeled objects. We created renders of these objects, and corresponding line drawings by detecting edges from the depth and normal maps of each render [68]. We then calculated image features for all renders and line drawings at each layer of Inception v3. The nearest neighbor accuracy of the line drawings was then computed with respect to the ImageNet features of the renders. When using features from the middle of the Inception v3 [75] network, the domain gap is reduced. We report nearest neighbor accuracies using features from the penultimate layer and the Mixed 6b node of Inception v3 in Table 8. When using the Mixed 6b node, the nearest neighbor accuracy for line drawings increases to 33.5% whereas the nearest neighbor accuracy using features from the penultimate layer is 24.5%. In contrast when we perform nearest neighbor classification on rendered images, the accuracy at the penultimate layer is 50.5% whereas the accuracy for earlier layers is lower at 38%.

Dataset	Accuracy (penultimate layer)	Accuracy (6b)
Line Drawings	24.5%	33.5%
Rendered Images	50.5%	38%
Random	2.5%	2.5%

Table 8. Reported nearest neighbor classification accuracies using ImageNet [18] features for various methods. The first column uses features from the penultimate layer and the last column uses intermediate features from earlier in the network at the Mixed 6b Node. The last row reports the probability of randomly picking the correct label.

Semantic Networks For the semantic constraint, we use the pretrained CLIP model with a vision transformer (ViT-B/32) base which is presented by OpenAI and is available under an MIT license [19, 66].

6.4. Datasets

Photograph Datasets Our training set for photographs comes from a randomly selected $10,000$ image subset of the Microsoft Common Objects in Context (COCO) dataset [53]. The COCO annotations and dataset fall under a Creative Commons 4.0 license, while the images were collected from Flickr and are subject to Flickr terms of use.

We use images from the MIT-Adoke 5k dataset for evaluation. This dataset is available for research purposes under its licenses.

Line Drawing Datasets The Contour Drawings dataset consists of $5,000$ images of boundary annotations for various scenes. We use the drawings with width 5. The dataset is available under a Creative Commons 3.0 license.

The Anime Sketch Colorization database consists of $14,224$ pairs of color drawings and sketches of anime char-

Figure 10. More results of our method in four different styles.

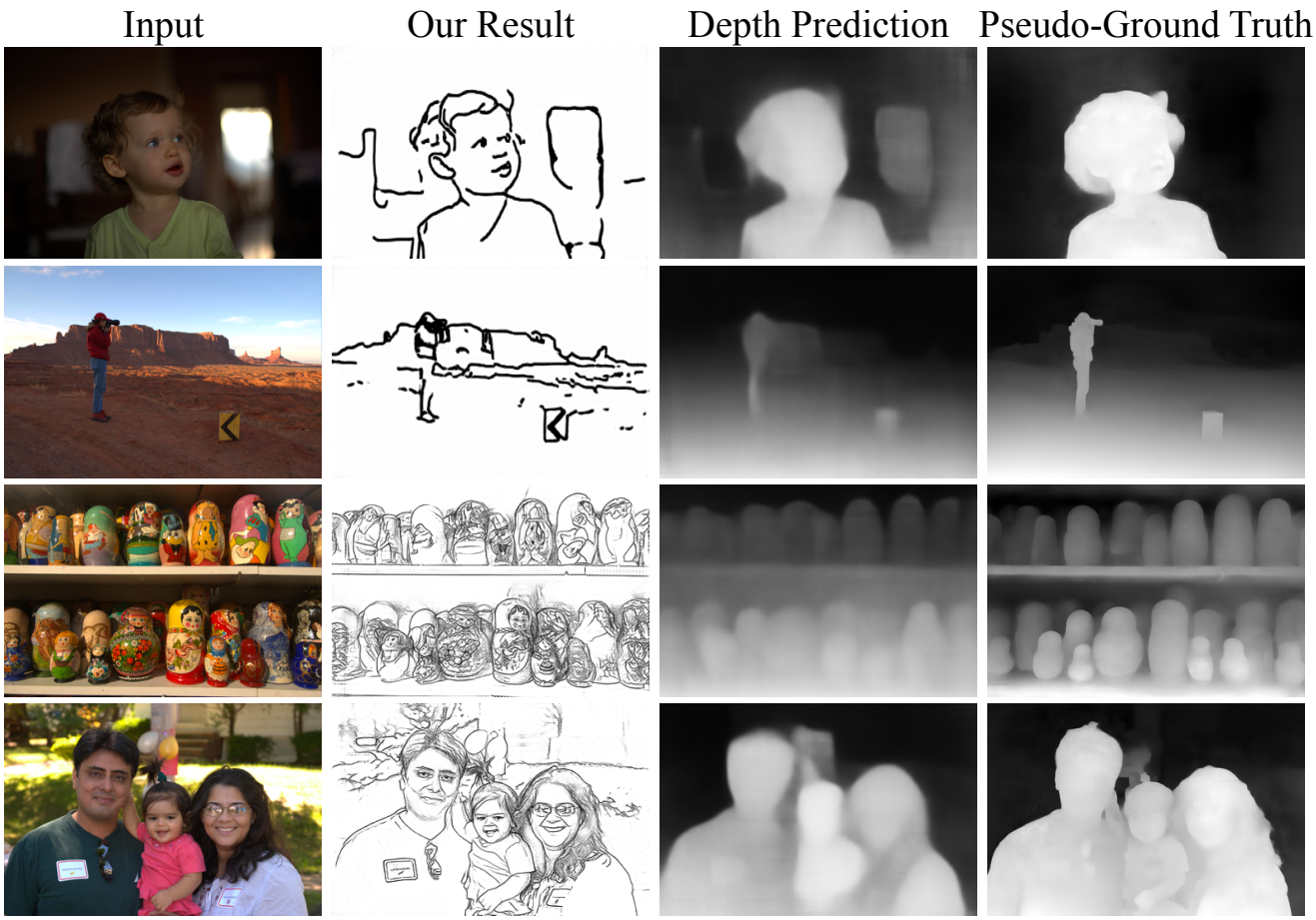

Figure 11. Depth predictions for our line drawings in two different styles. *Left to right:* Input photograph, our line drawing result, the predicted depth map from the line drawing from G_{Geom}, and the pseudo-ground truth depth map from pretrained network F.

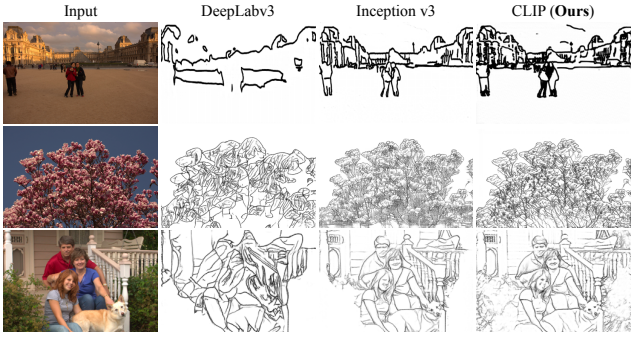

Figure 12. Different variants for the semantic loss. From left to right: the input image, output using DeepLabv3 features, output using Inception v3 features, and output using CLIP features (our model).

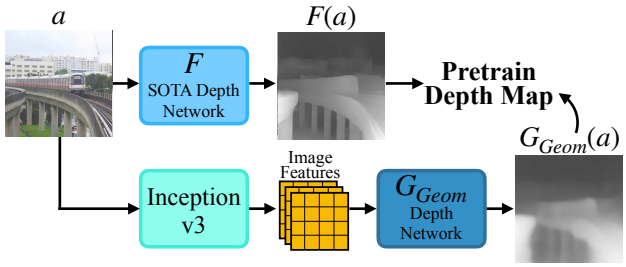

Figure 13. Network G_{Geom} is pretrained to predict depth maps from ImageNet features. G_{Geom} is supervised using state of the art depth prediction network F.

acters. In practice, we only use the sketches for the art style. This dataset is available under a Creative Commons 0 (CC0) license. As noted in the main paper, some images may contain sensitive content. For releasing our model, we selected 2256 training images without sensitive content.

OpenSketch consists of 420 product design sketches of 12 different objects. Sketches are drawn from multiple viewpoints by different artists. This dataset is available under a Creative Commons 0 (CC0) license.

We also use 207 artist sketches collected by Cole et al [15] to study where people draw lines. We could not find a license for this dataset although it is publicly available and provided by the authors.

Comparison	Total Judgements	Unique Users
Previous Work (Photo)	$10,000$	184
Previous Work (Portrait)	$4,000$	115
Ablation	$6,000$	90
Geometry Evaluation	$6,000$	82
Semantics Evaluation	$6,000$	135

Table 9. Number of total judgements and unique users for each perceptual study.

Portrait Datasets We use two main datasets for evaluating generated portrait drawings. The APDrawings Dataset [81] consists of 140 portrait, line drawing pairs for training (or 420 pairs after rotational augmentation) and 70 testing pairs. Although this dataset contains high quality images, it is limited in its size and all portraits are centered with similar poses and lighting. We could not find a license or information on data collection for this dataset, although it is publicly available and linked by the authors.

To create line drawings from portrait photographs under a large variation of poses, lighting, and general differences, we also evaluate our approach using the Helen Facial Feature Dataset [48]. This dataset contains 2000 training and 330 test high resolution photos collected for predicting facial annotations from a diverse range of photographs. Images in this dataset are collected from Flickr and subject to their own licenses and copyrights.

Since we did not have access to the full training set of both portraits and line drawings for style 1, we train our model using the Helen dataset and 88 line art portraits drawn by Charles Burns [6]. Although this second comparison is not exact, we wanted to include a scenario where our method matches the style of drawings created by UPDG.

6.5. Human subjects and User studies

Use use Amazon Mechanical Turk for all of our user studies. These studies were conducted with IRB approval. Before participating in the study, users were informed that if they consented, their responses could be presented in meetings and papers, and no personal information would be stored. Users were also told they could decline further participation at any time without adverse consequences.

For most comparisons, we gathered 1000 total judgements and users viewed up to 100 images. For comparisons on the APDrawings test dataset, we collected 1200 judgements and users viewed up to 70 images (the size of the test dataset). In Table 9 we report the total number of judgements and unique users for each study.