Learning to Draw Cats:
From Photos to Stylized Drawings

Authors: *Jingxi Qiu, Quan Yuan, Zhuoyan Guo*

6500 Final Project Report

Introduction

Many cat owners would love to have the hand-painted drawings of their pets, and some of them have the ambition to create the artwork of their pets themselves, as they feel that it is a meaningful experience. However, it is not always easy, especially for people with limited drawing skills. Thanks to advances in artificial intelligence and deep learning, there are a variety of tools that can transform a photo into a painting within a few seconds. However, most of the existing tools are trained on broader datasets, not specifically cat image datasets, and tend to generate cartoon or anime-style outputs. As a result, those tools frequently produce distorted or unrealistic images when applied to cat photos. Moreover, there are a limited number of AI tools that specialize in generating watercolor-style paintings because learning the subtle patterns and variety of human brush strokes is challenging for artificial intelligence.

Therefore, this project focuses on training a model specifically focused on generating watercolor-style cat paintings using Generative Adversarial Networks (GANs) using a cat image dataset in order to reach high fidelity, artistic authenticity, and cat-specific stylization. To achieve this, we experimented with multiple model architectures and workflows, including the CycleGAN model for unpaired image translation, a Multi-Stage pipeline for cat photo to anime Style conversion, and the Pix2Pix model for paired images.

Dataset

The cat photograph and watercolor painting data used in this project were manually collected from the social media platform *Red Note*. Specifically, two artist accounts – @lioo and @march – created original cat watercolor paintings and were used to build the dataset. In total, 245 paired images (one cat photo and a corresponding watercolor painting) were collected. Due to the time limitation, the dataset size remains relatively small compared to typical deep learning datasets, and this limits the following models' ability to generalize to unseen cat photos.

Figure 1. Example of paired training data: (left) real cat photograph; (right) watercolor painting.

Methodology

1.1 CycleGAN

1.1.1 Model Development

At the beginning of the project, the CycleGAN(*Zhu et al., 2017*) architecture was experimented with. CycleGAN is designed for unpaired image-to-image translation, and in this case, it is used to translate cat photographs to artistic-style drawings without needing exact photo-drawing pairs. This model includes two generators and two discriminators to train.

Noticeably, PatchGAN discriminator was used because of its characteristic that not just looks at the whole image but also focuses on small sections, which is helpful for the model to capture local details like fur texture and edges of the cat. Residual blocks were also employed as their features are ideal for retaining important information across layers and improving the overall image quality.

1.1.2 Model Results

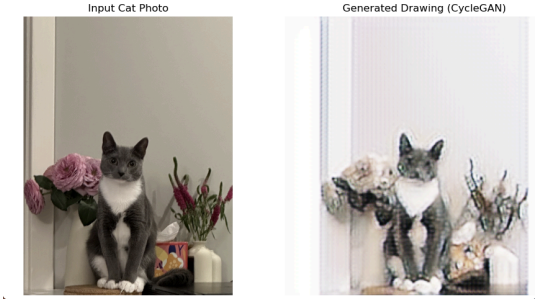

Figure 2. Unseen photos output generated by the CycleGAN model after 50 epochs of training

After training for 50 epochs, the generated outputs show the emergence of pink and pastel tones that provide a sketch or watercolor feel. However, the details of the cat are not clear and the reason could be that the dataset the drawings are also excluded the background and only focus on the cat itself, so the model may experience a hard time when the input cat photograph includes complex backgrounds.

2.1 Multi-Stage pipeline

Our approach draws inspiration from the traditional anime creation workflow employed by professional artists. Rather than attempting to transform photographs into anime-style images in a single step—a task that involves bridging substantially different visual domains—we developed a multi-stage pipeline that breaks this complex transformation into manageable sub-tasks. The pipeline consists of three stages.

2.1.1 Edge Detection Model

For stage 1, the edge detection, we implemented a custom neural network based on the VGG16 (*Theckedath & Sedamkar, 2020*) architecture. We designed a learning-based approach that could prioritize the essential structural elements. The model leverages the first 16 layers of a pre-trained VGG16 network as a feature extractor, followed by several convolutional layers that progressively reduce the feature dimensionality to produce a single-channel edge map. This architecture effectively combines the semantic understanding of a pre-trained network with specialized edge detection capabilities. Since obtaining hand-drawn edge maps as ground truth would be impractical at scale, we generated pseudo-ground truth using classical edge detection methods, which were then refined through targeted enhancements. We trained the model to minimize the binary cross-entropy loss between its predictions and these target edges.

2.1.2 Edge Detection Results

Figure 3. Input photo & output edge & input anime & output anime edge

The edge detection stage showed promising results in extracting structural information from cat photographs. Our custom neural network successfully captured key features such as facial contours, eyes, ears, and distinctive elements like bowties or accessories, while appropriately filtering out less relevant details like fur texture. The model performed particularly well on frontal cat faces with clear lighting conditions. However, we observed limitations when processing images with complex backgrounds, unusual poses, or poor lighting.

2.2.1 Line Refinement Model

For stage 2, the line refinement stage aims to transform photo-realistic edge maps into anime-style line art. For this purpose, we implemented a conditional Generative Adversarial Network (cGAN) based on the pix2pix framework. The encoder progressively reduces spatial dimensions while increasing feature channels, enabling the network to learn abstract representations. The decoder then reconstructs the image at its original resolution while incorporating the learned style characteristics. The discriminator employs a PatchGAN architecture, which evaluates the authenticity of overlapping image patches rather than the entire image. We trained the line refinement model using an adversarial approach combined with a reconstruction loss. Training alternated between updating the discriminator to better distinguish real and generated anime lines and updating the generator to produce more convincing anime-style line art. We employed the Adam optimizer with a learning rate of 0.0002 and trained for 50 epochs.

2.2.2 Line Refinement Model Results

Figure 4. Input photo & output edge & & output refined edge

The refined line output appears blurry and lacking definition compared to both the input edge maps and target anime lines. Important structural elements, such as eyes and facial features, often became indistinct in the refinement process. The model struggled to maintain the clear, decisive line quality characteristic of anime while attempting to simplify the input edges. These

results suggested that the line refinement stage represented a significant bottleneck in our pipeline, motivating our exploration of an alternative approach that bypassed this step.

2.3.1 Colorization Model

For stage 3, the colorization stage transforms anime-style line art into fully colored images. We implemented a deep residual network for this task, designed to learn the complex relationship between line structure and color distribution in anime. The residual connections help preserve structural information throughout the network, ensuring that the final colored image maintains the integrity of the input lines. We trained the colorization model using a combination of pixel-wise and perceptual losses. The primary loss function was the L1 distance between the generated colors and the target anime images, which encourages color accuracy. The model was trained for 50 epochs using the Adam optimizer with a learning rate of 0.0002, which was decreased gradually according to a cosine annealing schedule to ensure convergence.

2.3.2 Colorization Results

The colorization stage showed mixed results in transforming line art into fully colored anime-style images. When provided with high-quality anime-style line art, the model demonstrated a reasonable ability to apply appropriate colors, following the flat color aesthetic characteristic of anime. However, when processing the output from our line refinement stage, the colorization results were significantly compromised. The final images appeared blurry with low contrast and minimal definition. The model seemed to default to safe, neutral color choices rather than implementing the vibrant palette typical of anime.

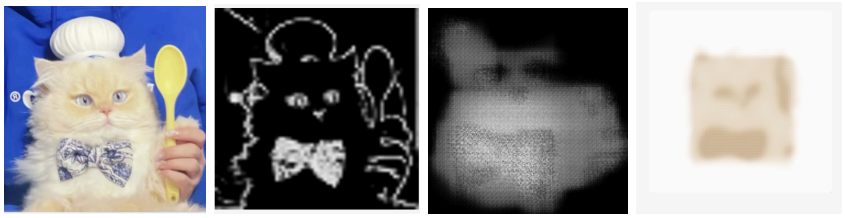

Figure 5. Input photo & output edge & & output refined edge & final result

3.1 Pix2Pix Model

3.1.1 Motivation

When each photo of a cat is precisely aligned with a hand-drawn rendition, a paired formulation is preferable to unpaired CycleGAN training. We adopt the conditional adversarial framework of Isola et al. (2017) to force the generator to learn a deterministic mapping from real cat image to a stylized image.

3.1.2 Network Architecture

Pix2Pix frames paired image translation as a conditional adversarial game given an input image x, the generator G must synthesize a corresponding output y (the drawing) that fools a discriminator D trained to distinguish real pairs (x, y) from fake pairs (x, G(x)).

Generator:

The generator adopts a classic UNet encoder-decoder: the encoder successively halves spatial resolution via stride 4×4 convolutions (feature depths 64 → 512), while the decoder mirrors this path with transposed-convolution up-sampling back to full resolution. Crucially, skip connections copy feature maps from each encoder stage to the matching decoder stage, allowing low-level edge and color cues to flow directly to later layers and preventing the "bottleneck blur" common in plain auto-encoders. A final tanh layer outputs a three-channel image scaled to $[-1,1]$.

Discriminator:

The discriminator is a lightweight PatchGAN: rather than classifying an entire image as real or fake, it slides a 70×70 receptive field across the concatenated input–output pair and produces a grid of realism scores. This design focuses learning capacity on high-frequency textures while keeping the model small (≈2.8 M parameters). Both networks are optimized jointly with a least-squares GAN loss for stable gradients, plus an L1 reconstruction term that anchors global correspondence between G(x) and the ground-truth drawing. Together, the UNet's spatial skip connections and PatchGAN's texture-sensitive critic form the core of the original Pix2Pix model.

3.1.3 Objective Function

$$L_G = \lambda_{GAN} L_{cGAN} + \lambda_1 \|\hat{y} - y\|_1 + \lambda_{perc} L_{perceptual}$$

L_{cGAN}: Conditional GAN, Least-Squares variant stabilises gradients.

$\|\hat{y} - y\|_1$: L1 reconstruction, Anchors global color and silhouette.

$L_{perceptual}$: Perceptual loss, VGG-19 features enforce edge & texture fidelity (see §3.2.2).

The model controls the generator training by minimizing these three losses to ensure the output contents, qualities, and details.

3.2 Methodological Enhancements

Because the training dataset is small and the manually extracted samples suffer from pixel misalignment, the model must impose substantial changes on the original images, making optimization particularly challenging. To address this, we propose several techniques designed to ease training and boost performance on small sample training.

3.2.1 Pixel-Level Image Alignment

To train a reliable Pix2Pix image-to-image translation model, each "input–target" pair must be pixel-aligned. Our alignment pipeline uses key-point detection to translate, rotate, and scale the real photo and its corresponding stylized image

We employ ORB (Oriented FAST and Rotated BRIEF) to detect feature points in both the real and stylized images, and additionally use Shi–Tomasi corner sampling together with dense-grid sampling to capture other structures (e.g., bow ties, spoon handles, whiskers), increasing the number of sampled points and improving matching performance.

The affine transformation is then estimated using RANSAC (re-projection threshold = 10 pixels), constraining the correction to translation, rotation, and uniform scaling. Finally, we warp the stylized image using warpAffine, applying constant white padding to eliminate border artifacts.

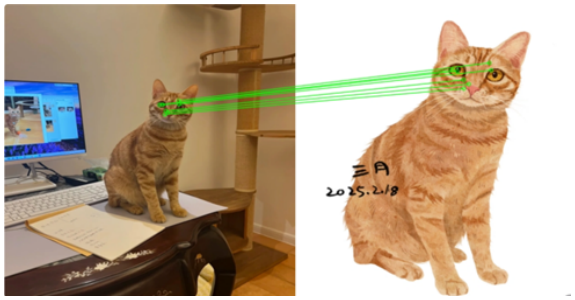

Figure 6. Key points between real and stylized image

Figure 7. The real image and the pixel-aligned stylized image

3.2.2 Perceptual Loss

Traditional Pix-to-Pix training minimizes a pixel-wise L1 distance between the generated image $\hat{y}$ and the ground-truth y. Although L1 preserves color fidelity, it penalizes every misaligned pixel, encouraging the generator to "copy–paste" input colors and producing overly smooth results.

We complement the L1 term with a perceptual loss (Johnson et al., 2016) that measures similarity in the latent space of a pretrained VGG-19 encoder. We then calculate the L1 loss of the embedding generated by the VGG-19 encoder from the real and generated images. Optimizing this objective shifts the inductive bias from raw RGB toward shape, edges, and mid-level texture, matching human perceptual preference in cat drawings.

3.2.3 Style Loss

To further disentangle structure from texture, we introduce the Gram-matrix style loss first proposed by Gatys et al (2016). The Gram matrix can be viewed as the centred covariance matrix of the features. In a feature map, each entry comes from the response of a specific filter at a specific spatial position, so every value represents the strength of a particular feature. Therefore, the Gram matrix captures the pairwise correlations between all features. With this

style representation in place, the stylistic difference between the two images can be measured simply by comparing the discrepancies between their respective Gram matrices.

3.2.4 Image transformation

By transforming and enhancing the data, we can expand the number of samples ten times, and improve the robustness to viewpoint changes and occlusion, forcing the model to predict pixels and preventing the model from copying pixels.

Transformation	Parameters	Rationale
Horizontal/vertical flip, random rotation	p=0.5, ±15	Inject viewpoint invariance
Random crop & resize	scale ∈ [0.8, 1.0]	Encourage localisation
ColorJitter + additive Gaussian noise	$\Delta hsv \leq 0.05$; $\sigma = 0.02$	Robustness to lighting/sensor noise
Random Erasing	p=0.5, area 2 – 10 %	Simulated occlusion forces global reasoning

Augmentations are applied consistently to paired samples to preserve alignment, for unpaired CycleGAN training, each domain is transformed independently.

3.2.5 Spatial Transformer Network (STN)

Hand-drawn targets often suffer from translations, anisotropic scaling, or mild rotations relative to input photos. We embedded a Spatial Transformer module (Jaderberg, 2015) in front of the UNet encoder to learn an affine grid that warps the incoming feature map.

It has a lightweight CNN predicting the six affine parameters, and a regular grid, and the Sampler performs bilinear resampling. The module is fully differentiable and incurs < 2 % computational overhead.

Results

4.1 Experimental Setup

Datasets – After pixel-level image alignment (§3.2.1), we have 187 valid images that can be used for image alignment for further training. We then split 169 images for training and 18 images for validation. The images are reshaped to 256*256 resolution and sent to the augmentation pipeline mentioned in 3.2.4.
Training details – The model is trained with a batch size of 8 for 500 epochs, 0.0002 learning rate schedule. For the loss rate, $\lambda_{l1} = 100$ and linearly drops to 5, $\lambda_{GAN} = 0.5$ and is 0 for the first 3 epochs in order to warmup, $\lambda_{perc} = 20$ and linearly drops to 5, $\lambda_{style} = 1$.

4.2 Metrics

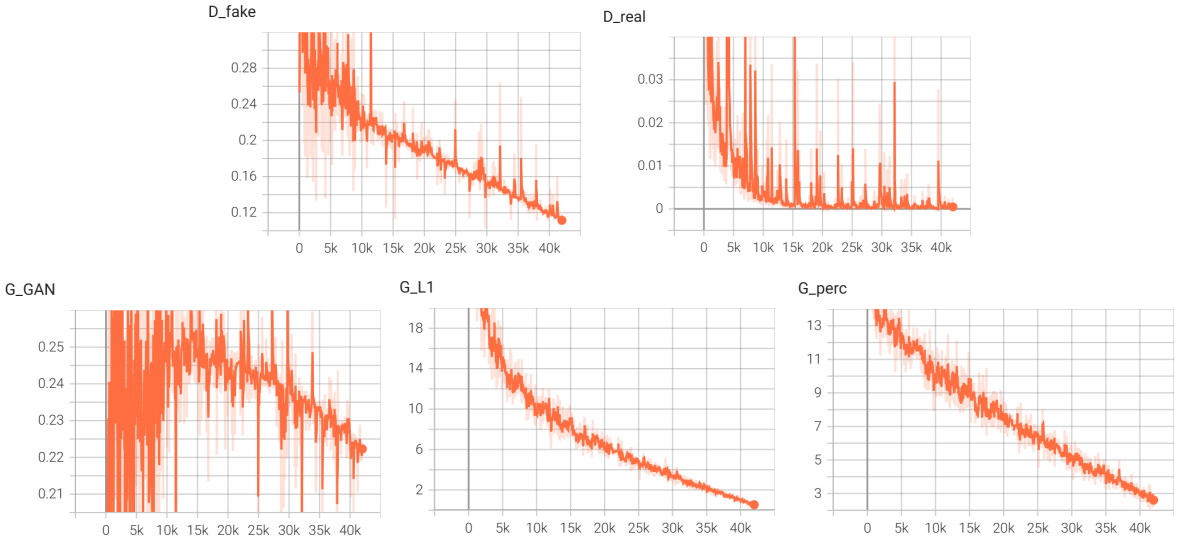

Figure 8. The Loss Curves of Discriminator and Generator

Figure 8 shows the evolution of the five losses that drive training. Both discriminator terms, D_real and D_fake, drop sharply within the first 10k iterations and then stabilize close to zero, indicating that the discriminator quickly learns to distinguish the real stylized image and the generated one, without causing mode collapse. The generator's adversarial term spikes briefly while the discriminator is warming up and subsequently oscillates within a tight band (≈ 0.22 – 0.24), signalling a numerically balanced min–max game.

By contrast, the pixel-wise reconstruction loss G_L1 falls from roughly 18 to less than 2, confirming that the network is steadily aligning global color and silhouette. And G_perc falls from 13 to 3 shows that the model has learned how to generate images with more texture information when training.

4.3 Result Visualization

Figure 9. The Real and Outputs of Cat Images

I. Single frontal body (first and second image)

When the input cat is located at the center of the image and is clear, the generator preserves the overall contour and reproduces the main color tone of the cat fur and surrounding props (such as the yellow blanket in the second column). The eyes and nose are still clearly distinguishable, indicating that as long as the source layout is similar to the samples seen during training, the network can transfer the mid-level structure.

II. Chaotic composition or multiple animals (third image).

The images of three cats exposed a key problem: the generator collapsed different faces into an amorphous mass and blurred the boundary edges. The model has not yet learned to focus on each head. And the locally unique evaluator of PatchGAN cannot enforce global instance consistency.

III. Significant attitude deviation or severe scale mismatch (fourth image).

Some rare perspectives and poorly aligned cropping in the training data can result in output underdrawing, leaving only blurry contours. This confirms the limitations mentioned in section 4.5: even small residual alignment deviations between photos and sketches can disrupt the pixel and perceptual losses of anchor training, resulting in rendering that is overly smooth or even almost blank.

Our model performs best on single-subject photos with pose and lighting conditions close to the training distribution and good center position, but performance sharply declines as composition complexity, scale variance, or alignment error increases. Addressing these weaknesses requires data enrichment (larger and more diverse pairing sets) and model upgrades (aligned perception modules, stronger global discriminators), as outlined in the future work roadmap.

4.4 Failure Modes & Error Analysis

Our qualitative inspection and metric trends expose these two major problems:

The first problem is with the blurred contours and fur texture, the UNet baseline tends to reproduce coarse color blocks while smoothing high-frequency details; this is aggravated by the dominant pixel-wise L1 term in early training.

The other problems are color bleeding and pastel halos. The local background picks up faint sketch pigments, a sign that the PatchGAN critic cannot fully police subtle texture boundaries when paired data are scarce.

The problems may be caused by three major factors:

Model: the vanilla UNet lacks global context modelling; receptive fields saturate before capturing long whiskers or ear tips. At the same time, because of the difficulty of the task, the

model is driven to achieve better indicators by copying and pasting pixels and blurring to get the average pixels.

Dataset: Only about 250 aligned pairs put the adversarial generation task in an overfitting state of constraint and amplification. Stylized pictures also come from multiple authors, with different styles, different alignment positions, and different decorative contents, which interfere with the model training. And because there are similar pictures in the data source (the same author creates the same cat from different angles), this may lead to some data leakage problems. The model performs well in images similar to the training sample, but poorly in images with insufficient data.

Task definition: Even in "paired" mode, hand-drawn targets are seldom pixel-perfectly aligned with their photos, violating the loss assumptions. This is too difficult for Pix2Pix's GAN model to learn.

4.5 Discussion & Future Work

Looking back at the steps and results of training the model, we believe that improvements can be made to the research in four major areas:

Data expansion & synthesis

Systematically crawl or crowdsource additional photo–drawing pairs, supplementing with unpaired sketches to unlock CycleGAN / diffusion training.

Leverage a high-quality, pre-trained generative model (e.g., DALL-E or SDXL fine-tuned on cats) to create synthetic drawings, increasing style diversity and learning samples.

Task redesign

Broaden the output domain beyond cats to multiple cartoon aesthetics, encouraging the model to learn style primitives rather than instance memorisation.

Model upgrades

Replace or augment UNet with globally-aware backbones: Vision Transformers, VQ-VAE encoders, or latent-diffusion decoders pretrained on large-scale animal imagery. We have found a library (PeterWang512/GANSketching) with pre-trained sketches for real cat generation. The weight of the pretraining model could be useful for transfer learning, as it already understands the concept of cats.

Investigate diffusion-based stylisation pipelines—Denoising Diffusion or ControlNet variants—whose iterative refinement is empirically sharper than single-shot GANs.

Introduce alignment-aware modules (such as Spatial Transformer) and multi-scale discriminators to better tolerate residual pose shifts. As we have introduced in class, we have utilized a ViT generator to catch information across the image; however, due to the constraints of decoder ability in such a difficult task, the attention mechanism hasn't been included.

References

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (ICCV) (pp. 2223–2232). https://doi.org/10.1109/ICCV.2017.244

Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, 1(2), 79.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14* (pp. 694-711). Springer International Publishing.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2414-2423).

Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. *Advances in neural information processing systems*, *28*.