CS230

Cat Photo-to-Sticker Translation with Deep Learning (Computer Vision, Generative Modeling)

Alan Lou
Stanford University
alanlou@stanford.edu

Cassie Zhang
Stanford University
hzhang20@stanford.edu

Elaine Liu
Stanford University
yilinliu@stanford.edu

Abstract

In this project, we aim to convert real cat images into watercolor-style cat stickers by utilizing image segmentation and image style transfer techniques. We used Mask R-CNN for the image segmentation task and used several different models, including neural style transfer, UNIT, CartoonGAN, CycleGAN and StyleGAN2, for the style transfer task. We performed both qualitative and quantitative evaluations for these models.

1 Introduction

Cats are one of the most popular pets in the world. Their companionship and emotional support have been important to countless people, especially during this pandemic lock-down period. We would like to build a model that transforms real-life cat images into cat stickers. It will not only be a fun and cute application for pet lovers, but also have commercial value for social media companies like Facebook and Snap.

The inputs for our model are real-life cat photos and the outputs are watercolor-style cat stickers. The cat sticker should preserve the original cat's identity while changing its style appearance.

2 Data Collection and Prepossessing

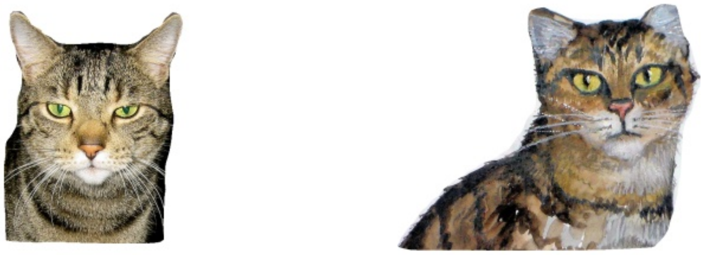
Our training set consists of both real cat images and watercolor cat images. We started by downloading images from Google Image and Bing Image. We then cropped out cats from the pictures to reduce background noise and make them more sticker-like. To do so, we ran Mask R-CNN [4] on real cat images using Detectron2 and pre-trained MS COCO model. Then we cropped the cats out using the detected masks. We also resized the images to $(256, 256)$ for faster computation later.

Since there is no pre-trained model to detect watercolor cats, we trained our own model to detect watercolor cats. We first created the training set by annotating outlines of 105 watercolor cats using VGG image annotator. We then performed transfer learning from the pre-trained weights of MS COCO model. Finally, we used the trained model to detect watercolor cats and cropped them out.

We realized there were significant variations in styles and quality of the watercolor images, so we manually went through the data set to remove low quality/inconsistent images and obtained around 1000 watercolor cat images.

3 Methods

Our model has two parts: segmenting and cropping cat images and performing style transfer.



(a) real cat image (b) watercolor cat image

Figure 1: Examples from dataset

The first part is straightforward. As mentioned in the previous section, we can apply Mask R-CNN to crop out the cats. Our main focus is on the second part - style transfer.

Neural style transfer method like [3] transfers the artistic style from a given reference to an image but does not exaggerate the geometry of the target object (cat in our case). There are several methods for unsupervised cross-domain image translation such as CycleGAN [16], UNIT [11], CartoonGAN [1] and StyleGAN [8], which suit our application better. We will compare these methods in the following sections.

3.1 Neural Style Transfer

Neural style transfer is a technique used to take two images — a content image and a style reference image - and blend them together so the output image looks like the content image, but in the style of the style reference image.

The performance of the neural style transfer method depends heavily on the style reference image. After exploring different reference images, we picked the one that generates the best blended images.

The advantage of this method is that it is easy to implement and we can influence the style by choosing different style reference images. The drawback of this method is that it does not alter the geometry of the target object, and we do not have fine-grained level control of the output features.

We used the implementation of Neural Style Transfer by titu1994 [12] to perform the style transfer.

3.2 UNIT

UNsupervised image-to-image translation (UNIT) aims at learning a joint distribution of images in different domains by using images from the marginal distributions in individual domains. The authors of UNIT made a shared-latent space assumption and proposed an unsupervised image-to-image translation framework based on Coupled GANs. And the shared latent space constraint implies cycle-consistency constraint. [11] In our case, real cats and watercolor cats are the two domains here. There are six sub-networks: two image encoders for both domains E1 and E2, two image generators G1 and G2, and two adversarial discriminators D1 and D2. We used the UNIT Tensorflow implementation by Junho Kim [10] and trained 170 epochs in Google Colab using 1 GPU.

The trained model performs quite well on frontal, close-up head shots with good lighting. In these cases, the model is able to generate artistic cat stickers that vividly capture colors and facial features of the cats. However, it performs poorly on full body pictures: the model cannot recognize the cats' body parts correctly and sometimes generates a cloud of mixed colors without a distinct cat face. This might be caused by the lack of full body watercolor cat images in the training set because most of the cat painting online are frontal close-up portraits. It is possible that the model is unable to map full body cat images to accurate latent code in the shared latent space to capture different body parts.

3.3 CartoonGAN

CartoonGAN is a GAN that focuses specifically on transforming real world images into cartoon-style images. The model was trained on unpaired real-world photos and cartoon images to find a mapping from photo domain to cartoon domain. We explored transforming real cat photo to cartoon cat photo using CartoonGAN pretrained on Shinkai Makoto anime style.

The transformed cat cartoons look reasonable; however, the model is only capable of transforming to anime styles and is not able to perform geometric transformation particular to cats.

3.4 CycleGAN

CycleGAN is a GAN similar to CartoonGAN that performs translation of images from source domain to target domain and is trained on unpaired images from two domains. One thing special about CycleGAN is that it also introduces cycle-consistency loss which constrains the transformed image to be close to the original image. One popular application of CycleGAN is translating images of horses to zebras. It's worth noting that on CycleGAN's official Github page, the author noted that CycleGAN does not perform well for tasks that require geometric changes.

Since there is no pretrained cat to cartoon cat CycleGAN model, we trained our own model using the CycleGAN tensorflow implementation[5] developed by Zhenliang He. The model was trained on 1400 cropped real cat images and 970 cropped watercolor cat images. The trained model is able to transform some of the test cat images into watercolor styles relatively well; however, the model seems to perform worse on cat images that are of low resolution or low brightness.

Another interesting observation is, compared to output images of other models, the ones outputted by CycleGAN have thicker "borders" around the cats that look like blended watercolor. We speculated that this could be because CycleGAN includes a cycle-consistency loss term, and adding the watercolor blend around cats is an easier way to pass the discriminator test as well as transform the picture back to real cat image.

3.5 StyleGAN2

StyleGAN is one of the more recent GANs that automatically learns and separates high-level attributes and stochastic variation in generated images. Inspired by the ideas behind website "Toonify Yourself" [14], we developed a process that blends two StyleGAN2 models to accomplish the task of image to image translation.

Firstly, we trained a base model using the cropped real cat images. Then, we trained a fine-tuned model using transfer learning and fine-tuned on watercolor cat images. To perform the style transfer, we swap lower resolution layers from the base model into the fine-tuned model, which serves to preserve the cat's original posture while transferring features and style of the watercolor cat.

To generate a watercolor cat image from an arbitrary cat image, we use back-propagation to find the latent vector in the base real cat StyleGAN2 model that will produce an output looks almost like the real cat, and then we pass that latent vector to the blended model described above.

The advantage of this method is that the output image is of high-resolution. It also allows for superior control and understanding of generated images. The disadvantage of this method is that some features may shift in the fine-tuning process. For example, the eye color may change from blue to black after fine-tuning because black-eye cats are more commonly presented in the training dataset. As a result, sometimes the generated watercolor cat may look different from the original cat if we compare them based on specific features.

For StyleGAN2 model fine-tuning, we used the code provided in the NVIDIA's StyleGAN2 project [9]. For model blending and latent vector projection, we used a forked version of NVIDIA's StyleGAN2 project implemented by justinpinkney [7].

4 Results and Evaluation

We applied different model on a test set and present a visual comparison in Figure 2. Comparatively speaking, Neural Style Transfer, StyleGAN and CartoonGAN generated the most consistent results

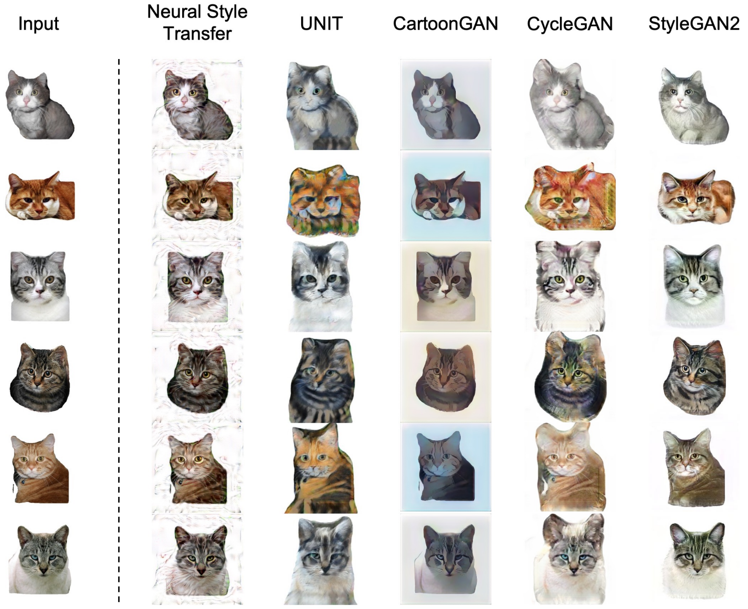


Figure 2: Different methods for real cat to watercolor cat image translation. From left to right: input, neural style transfer [3], UNIT [11], CartoonGAN [1], CycleGAN [16], and StyleGAN2 [8].

and are more "realistic". UNIT seems to be more artistic, but the quality varies. CycleGAN is able to preserve the original cat features and capture watercolor stylistic features, but it always has a "rim" around the cat.

To evaluate the models' performances quantitatively, we calculated the FID (Fréchet Inception Distance) and conducted perceptual study. We focused the evaluations on models trained/evaluated on the watercolor cat image data set (UNIT, StyleGAN, CycleGAN, Neural Transfer), since these models are more comparable with one another. CartoonGAN is omitted in the comparison below because it is providing a more cartoon style transformation rather than the watercolor style.

4.1 FID

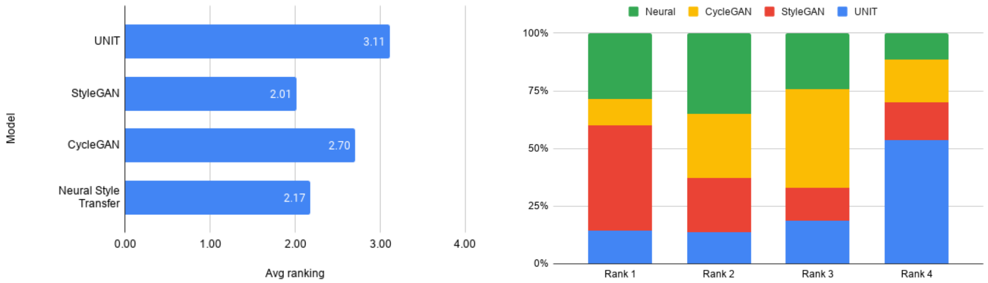
FID (short for Fréchet Inception Distance) is a metric that summarizes how "similar" two sets of images are, based on the distance between activations of some layer of the inception v3 model, using input images from the two datasets [2]. Lower FID indicates more similarity between datasets. Note that inception v3 model was trained on the ImageNet dataset, which contains only images of real world objects, thus FID measured on watercolor cat images may not be as good a reference as FID measured on real cat images.

For each model, we calculated the FID between 200 model output images and 200 real cat images, as well as with 200 watercolor cat images using an open source PyTorch implementation[13], using the second max pooling features. The results are listed in Table 1.

Model	FID with real cat	FID with watercolor cat
UNIT	14.00	6.40
StyleGAN	2.52	8.79
CycleGAN	18.97	4.54
Neural Transfer	61.05	28.10

Table 1: FID results

From the FID results, we found that images generated by StyleGAN is most similar to real cat images, and images generated by CycleGAN is most similar to watercolor cat images. UNIT's performance



(a) Average model rankings

(b) Percentage of each model in the 4 ranks

Figure 3: Perceptual study results

is in between StyleGAN and CycleGAN. Neural transfer has highest FID with both real cat images and watercolor cat images.

4.2 Perceptual Study

The perceptual study assess the visual quality of stickers generated by different models. We presented 10 sets of images to 14 participants. Each set includes the original cat pictures and four images generated by Neural Style Transfer, CycleGAN, StyleGAN2, and UNIT (with order randomized in each set). We asked the participants to rank their favorite cat image to sticker transformation from 1-4 for each set of images.

We summarized the study results in Figure 3. The first chart shows the average ranking for each model: StyleGAN achieved the highest average ranking of 2.01, Neural Style Transfer ranked second closely after StyleGAN. The second chart shows the percentage of each model in different ranks. Each rank consists of all four models. It is evident that participants showed strong personal artistic preferences for the models. Some participants cited they prefer more realistic stickers, and others suggested they appreciate more artistic transformations.

While we conclude that StyleGAN performed best in this perceptual study, we also acknowledge other models have their own strengths and may be more aesthetically pleasing for certain people. In real world applications, we could offer users different choices of cat stickers generated by different models, and have them pick their favorite ones.

5 Future work

Since the performance of the StyleGAN2 model looks promising, we can publish a web application for users to create their own cat stickers. However, finding the latent representation of cats in the StyleGAN2 model is pretty expensive and is unlikely to make it as a web app. To speed up the process, we can replace the expensive optimisation process by distilling the combined StyleGAN2 generator into an image-to-image network trained in paired way [15]. In other words, we can train a pix2pixHD model to apply the transformation to any arbitrary cat image, which will bypass the overhead of the optimisation step.

We are also planning to implement Multimodal UNsupervised Image-to-image Translation. While UNIT assumes a fully shared latent space between two domains, MUNIT assumes that only part of the latent space (the content) can be shared across domains whereas the style is domain specific, which could lead to better results for our case [6].

We would also like to collect more training set to include different styles. For example, we can apply the same process to generate cartoon-style cat stickers. The advantages of GANs may be more obvious when the target style diverges further away from real cat images.

6 Contributions

All team members contributed equally to this project.

Alan worked on collecting and cropping images, training neural style transfer model, and training StyleGAN2 model. Cassie worked on training CartoonGAN model, training CycleGAN model, and calculating FID. Elaine worked on training Mask R-CNN for watercolor cats, training UNIT model, and conducting the perceptual study. All team members worked on literature research and writing the reports.

Code for this project and sample outputs are available at: `https://github.com/alanlou/cs230_project`

References

[1] Filip Andersson and Simon Arvidsson. *Generative Adversarial Networks for photo to Hayao Miyazaki style cartoons*. 2020. arXiv: `2005.07702 [cs.GR]`.

[2] Jason Brownlee. *How to Implement the Frechet Inception Distance (FID) for Evaluating GANs*. `https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/`. 2019.

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "A Neural Algorithm of Artistic Style". In: *CoRR* abs/1508.06576 (2015). arXiv: `1508.06576`. URL: `http://arxiv.org/abs/1508.06576`.

[4] Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: `1703.06870`. URL: `http://arxiv.org/abs/1703.06870`.

[5] Zhenliang He. *CycleGAN-Tensorflow-2*. `https://github.com/https://github.com/LynnHo/CycleGAN-Tensorflow-2`. 2018.

[6] Xun Huang et al. "Multimodal Unsupervised Image-to-Image Translation". In: *CoRR* abs/1804.04732 (2018). arXiv: `1804.04732`. URL: `http://arxiv.org/abs/1804.04732`.

[7] justinpinkney. *stylegan2*. `https://github.com/justinpinkney/stylegan2`. 2020.

[8] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *CoRR* abs/1812.04948 (2018). arXiv: `1812.04948`. URL: `http://arxiv.org/abs/1812.04948`.

[9] Tero Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *Proc. CVPR*. 2020.

[10] Junho Kim. *UNIT-Tensorflow*. `https://github.com/taki0112/UNIT-Tensorflow`. 2017.

[11] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. *Unsupervised Image-to-Image Translation Networks*. 2018. arXiv: `1703.00848 [cs.CV]`.

[12] Somshubra Majumdar. *Neural-Style-Transfer*. `https://github.com/titu1994/Neural-Style-Transfer`. 2019.

[13] mseitzer. *pytorch-fid*. `https://github.com/mseitzer/pytorch-fid`. 2017.

[14] Justin Pinkney. 2020. URL: `https://toonify.justinpinkney.com/`.

[15] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. *StyleGAN2 Distillation for Feed-forward Image Manipulation*. 2020. arXiv: `2003.03581 [cs.CV]`.

[16] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2020. arXiv: `1703.10593 [cs.CV]`.