Automatic Manga Colorization with Hint

Junjie Ke, Honghao Wei, Yiwei Zhao

CS231N Final Project

Stanford University

Stanford | ENGINEERING
Computer Science

Abstract

We investigate different methods to **generate colorful cartoon images** from black and white sketches, together with the color hints given by user. An end-to-end method is firstly implemented using CNN with direct links, namely **uNet**. We also try conditional generative adversarial networks (**cGAN**) [1] [2], Wasserstein GAN **(WGAN)** [3] and **improved WGAN** [4] to improve the generating quality.

Problem Statement

Problem:
Given color hint and line art image, we colorize the sketch.
Dataset:
20000 colored manga images from safebooru.org

Method

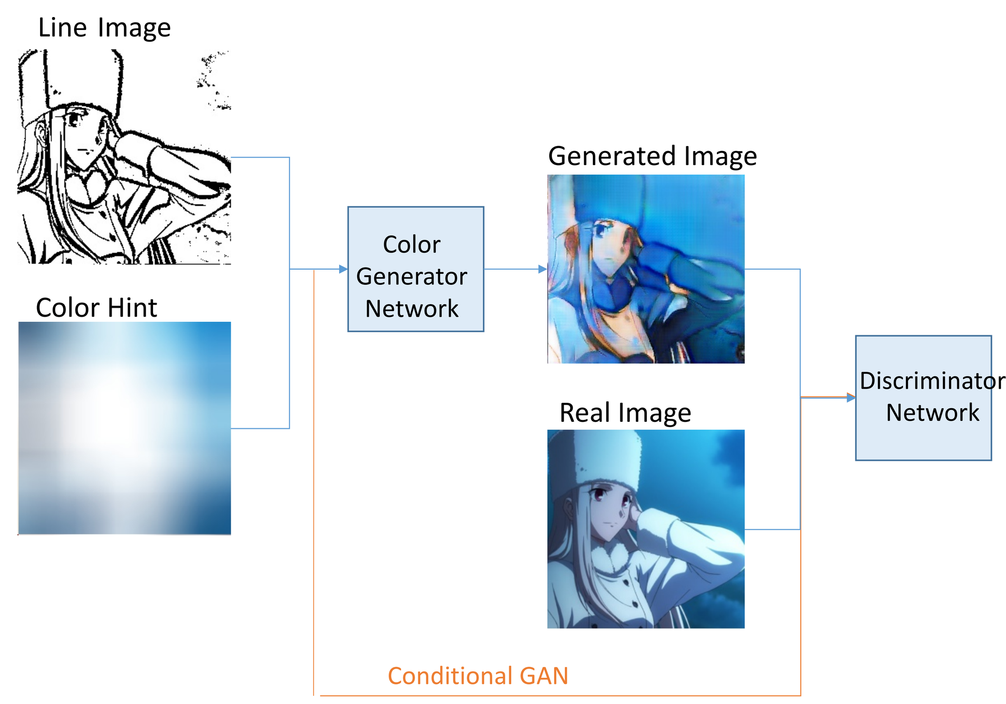

Fig 1. GAN System for Learning to Colorize Manga

Line image: We use OpenCV to detect the boundary of the image and extract the sketches from colored image.
Color hint: In addition to the line image, we'll give the network another image containing the colors of the original image. During training of cGAN, the line image and color hint are also fed to the discriminator network.

Wasserstein GAN value function:

$$\min_G \max_{D\in\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}}))]$$

Improved-WGAN training strategy: We follow [4] to use gradient penalty to enforce the Lipschitz constraint. The objective function is as follows:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

L1 loss: Previous approaches of c-GANs [2] have found it beneficial to mix the GAN objective with a more conventional loss functions. In this paper, we also use the L1 distance to describe the pixel-level loss in our model.
VGG feature map: We also employ a pre-trained VGG19 to extract high-level information of the image. We extract the outputs of the final convolution layer as feature map and compute the **L2 distance** as feature loss.
Evaluation metrics: The Wasserstein distance **(W-distance)** between real and generated data provides a useful metric of convergence [3], it can be approximated by **(-1 * d_loss)**. Lower W-distance would correspond to higher quality images.

Network Structure

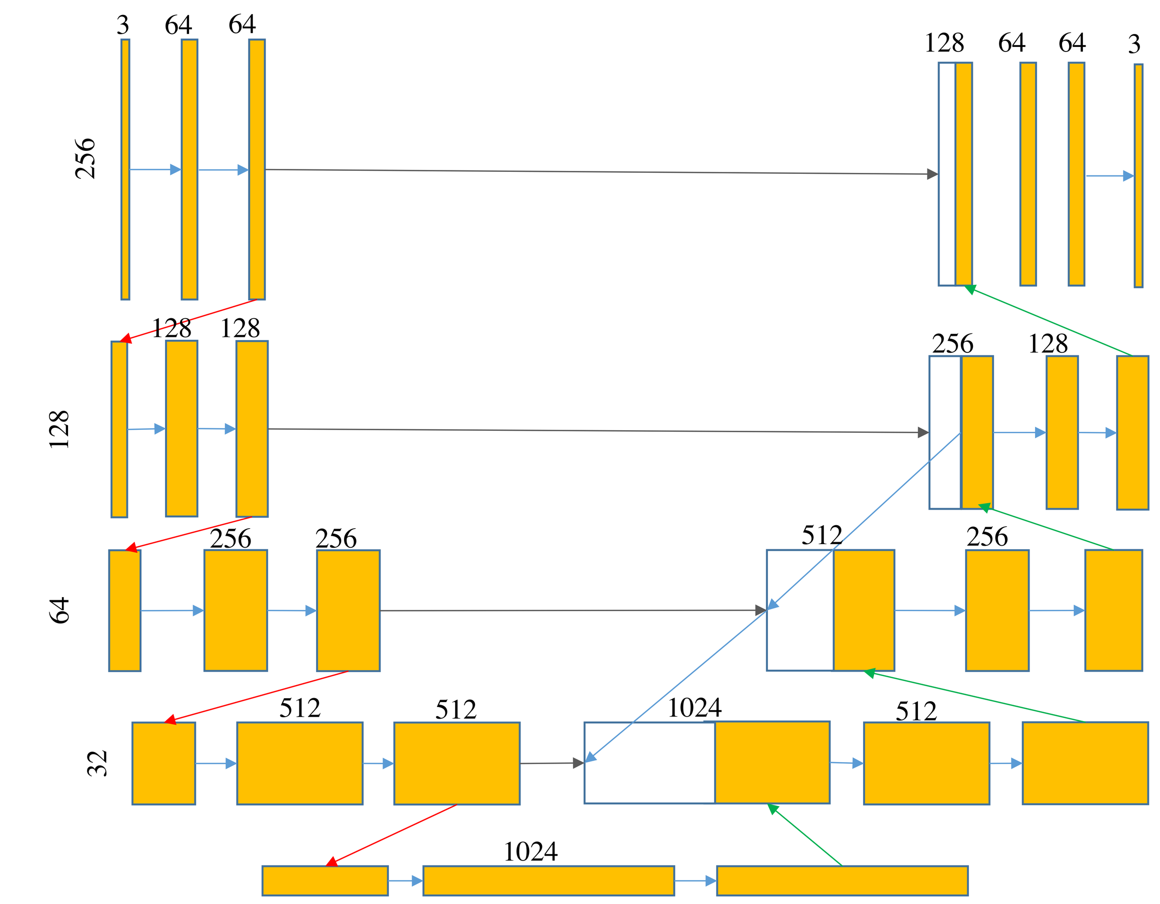

Fig 2. Generator Network Structure

Generator: The generator produces a colored image based on line image and color hints. Instead of the encoder-decorder structure, we employ the "U-Net"[5] by concatenating layers in encoder to the corresponding layers of the decoder. The network structure is in Fig 2.

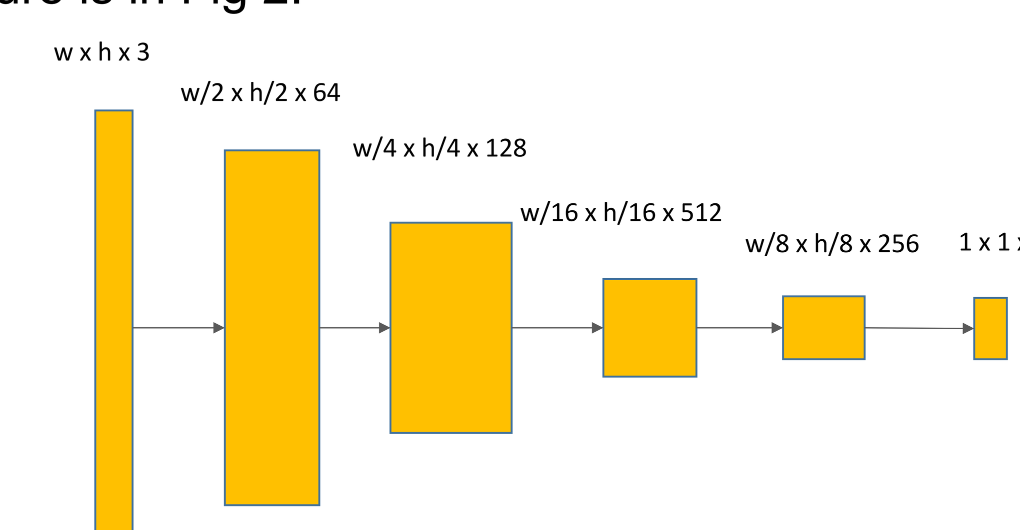

Fig 3. Discriminator Network Structure

Discriminator: The discriminator compares the generated image with the real image. Its input is the concatenation of line image, color hints and the generated/real image. In our network, we use a simple stack of convolutional layers to output a probability scalar. The network structure is in Fig 3.

Quantitative Results

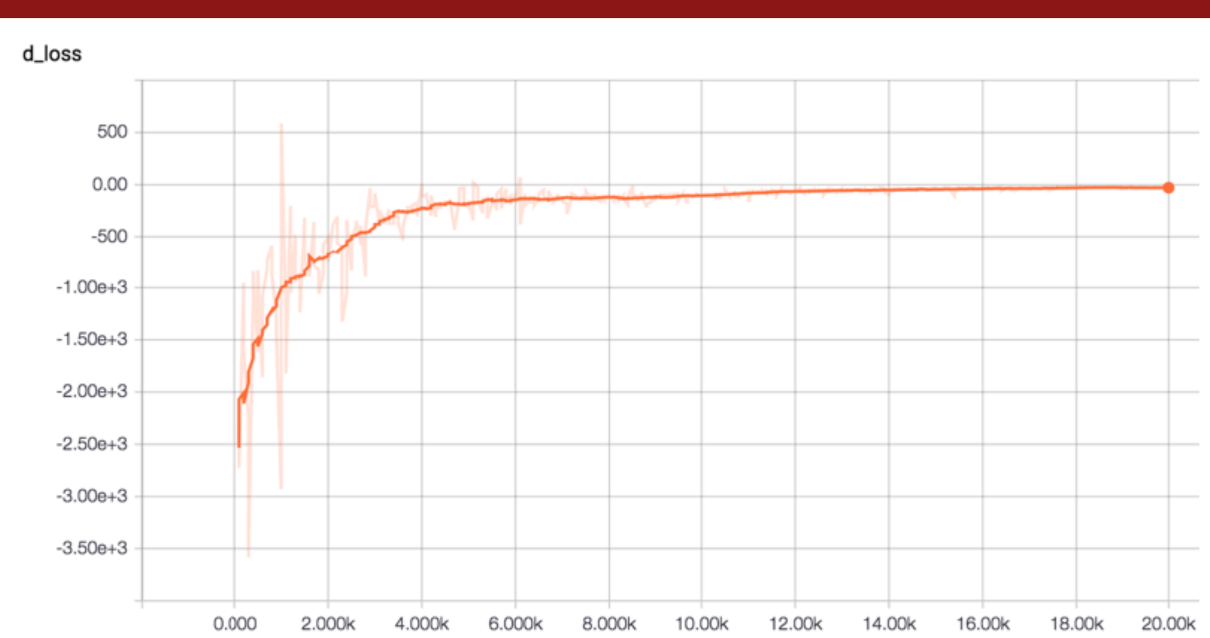

Fig 4. Improved-WGAN discriminator loss curve.

(-1 * d_loss) represents the **W-distance**. Smaller the distance represents higher similarity between the generated images and real images. From Fig 4 we can see that the discriminator's loss **steadily go up until convergence.**

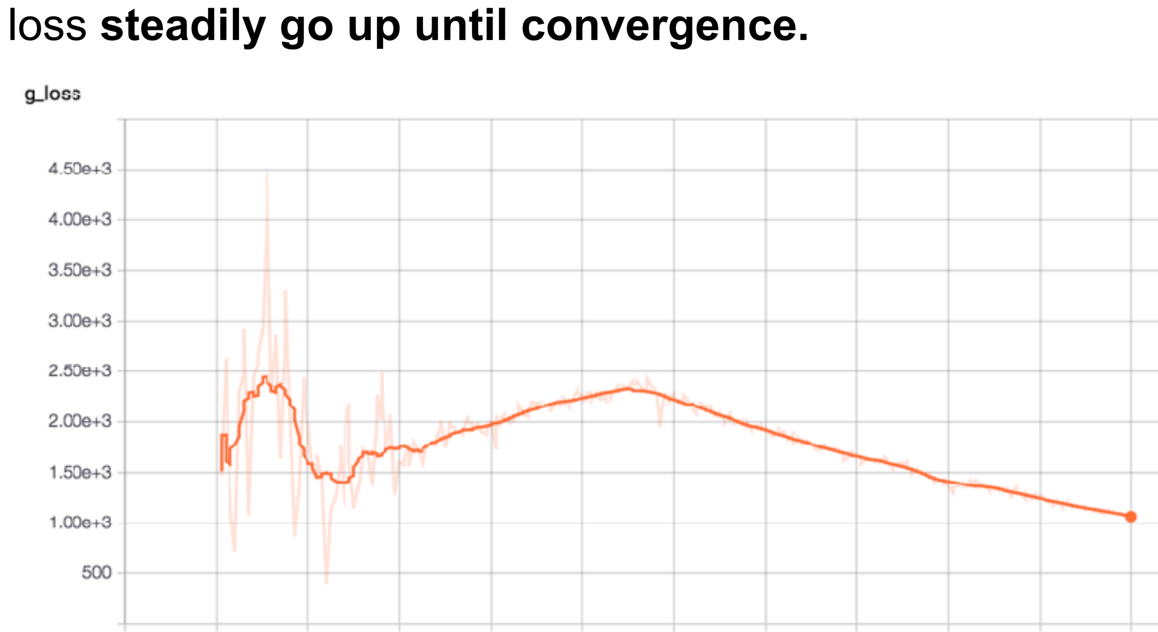

Fig 5. Improved-WGAN Generator Loss curve.

Experiment Results

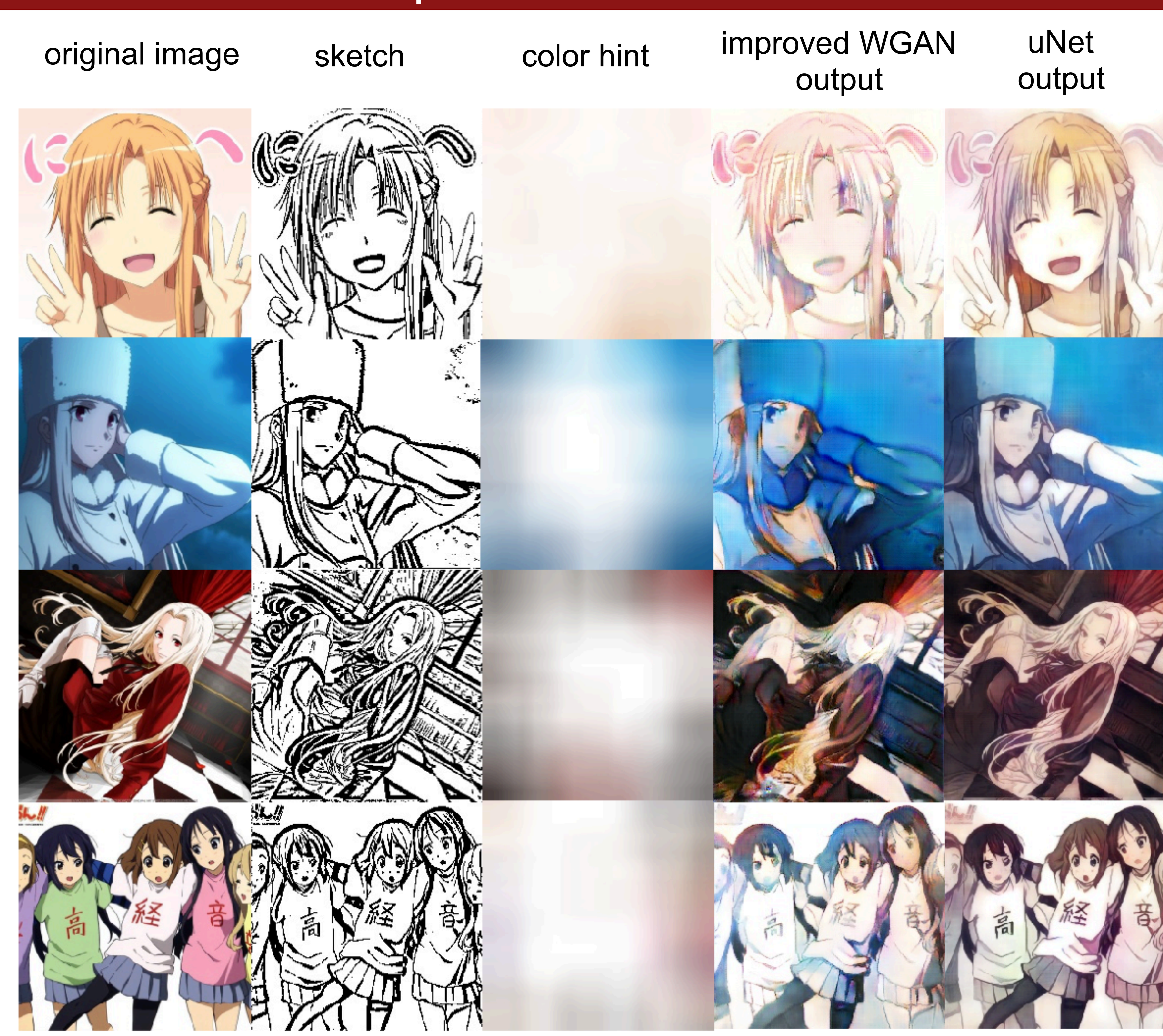

Fig 6. Experiment Results on Test Set

References

[1] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.
[2] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).
[3] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875* (2017).
[4] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." *arXiv preprint arXiv:1704.00028* (2017).
[5] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer International Publishing, 2015.
[6] Liu, Yifan, et al. "Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks." *arXiv preprint arXiv:1705.01908* (2017).