Anime Sketch Colorization

Title: Create an anime sketch colorization model using DNN, GAN and Swish-gated Residual U-Net.

Abstract: Nowadays anime sketch colouring has become a new research hotspot in the field of deep learning. In this task aim is to build a Deep Neural network model using GAN and swish-gated Residual U-Net to produce colour anime images from given black and white sketches. In this task to fill the black and white sketch with a variety of colours to make the image colourful.

Task:To build a model a machine learning model which can create a color anime Image from the given black and which anime sketch image.

Keywords: Deep Learning, GAN, U-Net, swish-gated Residual U-Net.

Problem Link: https://www.kaggle.com/wuhecong/danbooru-sketch-pair-128x

Dataset : Danbooru sketch images with 128x
https://www.kaggle.com/wuhecong/danbooru-sketch-pair-128x

Introduction:

In the field of anime drawing, people usually draw an anime sketch first. It is a time consuming process to colour the sketch image. This part takes more time. Is there any solution for this problem?

Recently, Generative Adversarial Networks (GANs) have been used for anime sketch colouring and some anime sketch colouring models based on GANs are proposed.

GANs[1] are composed of Generator and Discriminator models. Both are multi-layer perceptron (MLP) models. It learns to generate new data with the same statistics as the training set. In anime sketch colorization Generator is used to generate colourful anime images and discriminator is used to check whether generated image is 'real' or 'fake'. Network train to maximize classification accuracy.

Objective function of the GAN .

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

It is a min max problem. Where *x* is the original input, z is random noise. D is a Discriminator model and G is generator model.

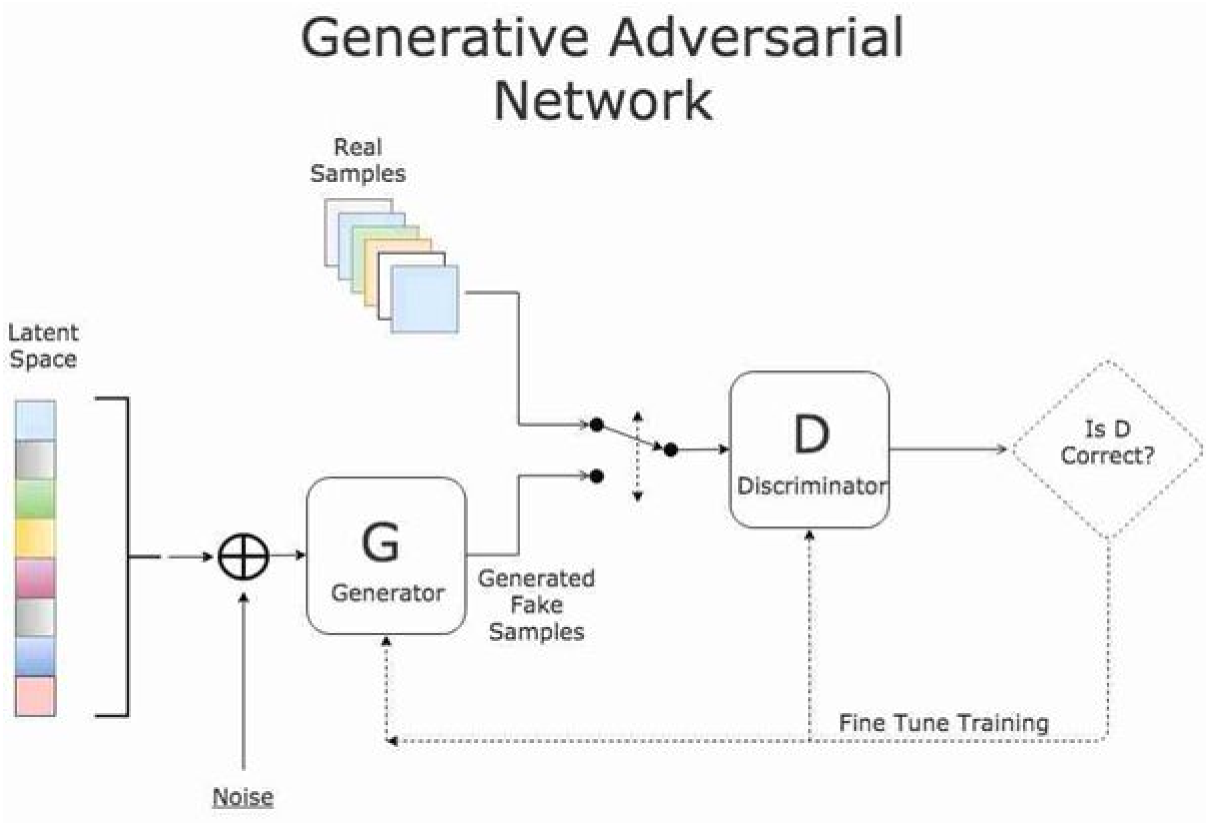


Fig: GAN (source: https://mc.ai/keras-optimizers-comparison-on-gan/_)

In GAN Generator and discriminator are DNNs.

Training a GAN:

We can simultaneously train Generator and Discriminator on batch data:

1. Generator: It is a DNN model which generates noisy data . Which generates new data.

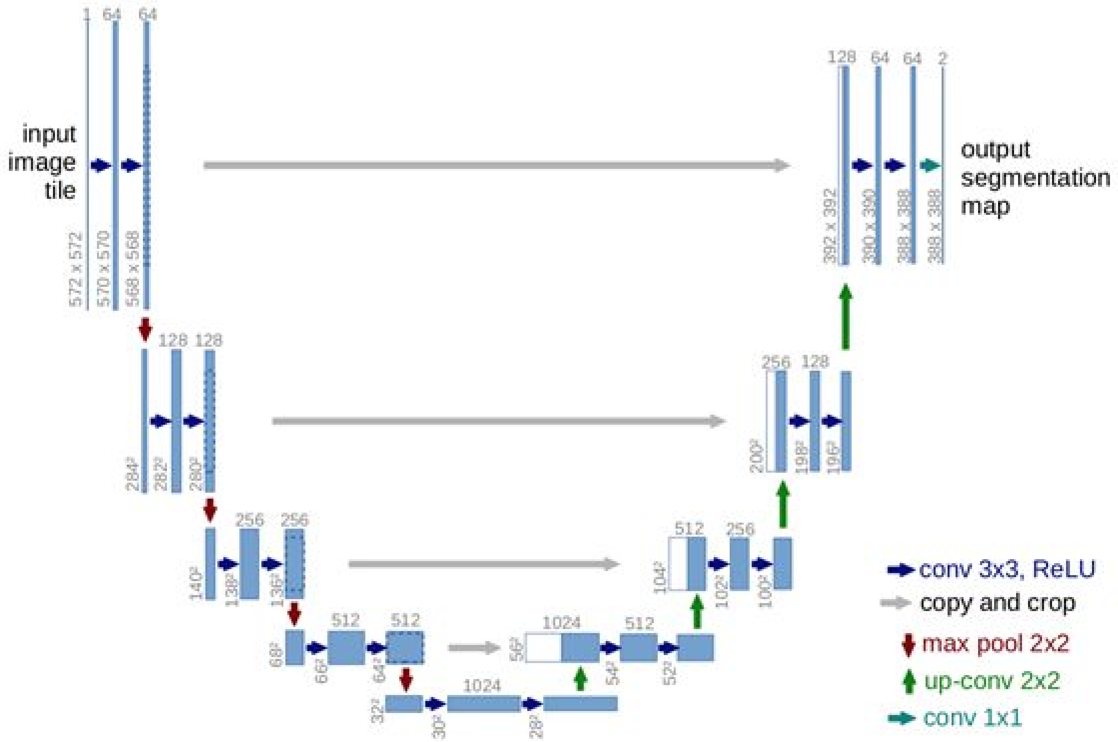
2. Discriminator: It is also [0 – 1] classifier model. It has both real images and generators created fake images as input. For train discriminators provide real image data as class label 1 and fake data with class label 0. So it can distinguish between real and fake images(Generated images).

3. After the discriminator is trained we generate fake images with class label 1. Now test that images on the GAN model by making Discriminator trainable false. Which only train generator model. Find

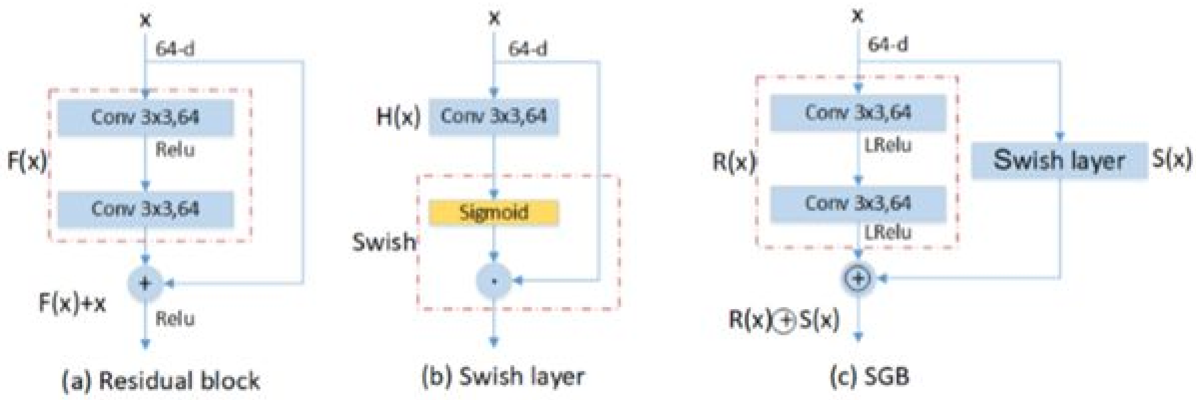
out loss using sigmoid function. It determines the generated image is close to the manual colouring image.

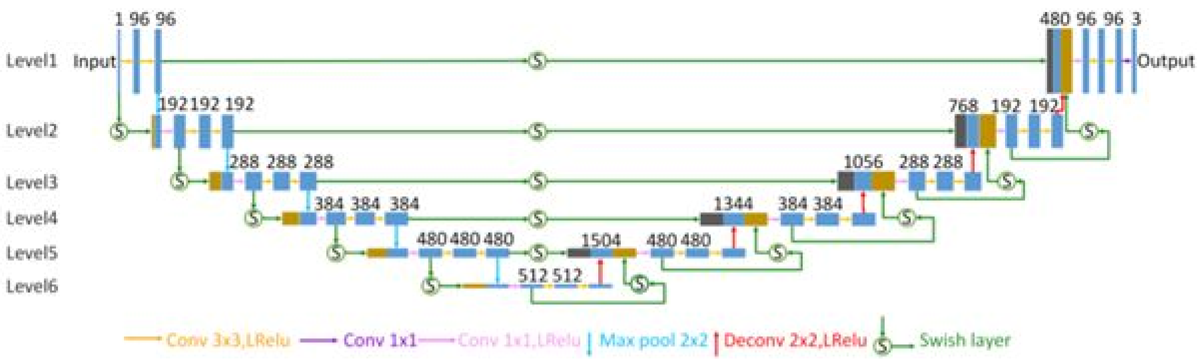
Many colouring GAN models use U-Net as the generator model.

U-Net: It is a network encoder-decoder structure. It is often used in the image segmentation. U-Net is composed of two similar branches that can extract the feature from the images. It consists of image to image transfer task. Fig shows the basic architecture of U-Net [9].



But using simple U-Net has a problem of poor colouring image, unreasonable colour mixing, dramatic changes in colour brightness, colouring beyond the filled areas and so on. To overcome this problem use **switch gated residual U-Net**. To improve the U-net as increase the depth of the network makes it stable.





Literature Survey:

Many researchers have proposed many GANs variants for improvement GAN.

Liu [2] proposed model Auto-painter Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks. They proposed the Conditional GAN. It can automatically produce color images from a given sketch. It also allows users to indicate preferred colour.

In these they used conditional GANs. GANs are unconditioned generators where generator model mapping from the random noise data vector *z* to output *y*: G: z -> y. (output is generated using random vector z). In conditional GAN output depends on the {x, z} x is input to the generator and z is random noise {x, z} -> y. In this problem x is the sketch image.

Loss Function: Instead of a simple encoder decoder model they used U-net. It is a pix2pix model. The objective(loss function) cGANs.

$$\min_G \max_D V(G,D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x), z\sim p_{data}(z)}[\log(1 - D(x, G(x,z)))]$$

In this G – Generator Model output D- Discriminator Model output. X is black and white sketch image and y(target) is a color image, z is random noise. In this paper they also use **pixel-level loss** i.e Lp is L1 distance between pixel to pixel. Lp is the difference between generated image and original image.

$$L_p = \mathbb{E}_{x,y\sim p_{data(x,y)}, z\sim p_{data}(z)}\left[\|y - G(x,z)\|_1\right]$$

They used a VGG-16 net to extract high-level information of image data.

$$L_f = \mathbb{E}_{x,y\sim p_{data(x,y)}, z\sim p_{data}(z)}\left[\|\phi_j(y) - \phi_j(G(x,z))\|_2\right]$$

Lf is L2 between feature spaces. Φj is the j^{th} layer VGG16 output of image. Find out the difference in generated and original image.

In [3] they proposed residual U-Net to apply the style to the grayscale sketch with auxiliary classifier generative adversarial network (AC-GAN).

This paper proposed adding two guide decoders at the entrance and exit of the residual u-net intermediate layer and input the reference style image to VGG16/19 to get the fc1 layer 4096 output and then be as the global style to the middle layer of the residual u-net.

Loss function is:

$$L_{l1}(V, G_{f,g_1,g_2}) = \mathbb{E}_{x,y\sim P_{data}(x,y)}[||y - G_f(x, V(x))||_1 + \alpha||y - G_{g_1}(x)||_1 + \beta||y - G_{g_2}(x, V(x))||_1]$$

V(x) is the output of VGG19, G_f final output of U-Net, G_{g1} decoder at entry and G_{g2} decoder at middle exit.

In Gang Liu[4] they proposed Swish-gated Residual U-net and Spectrally Normalized GAN (SGRU), a method for automatic colorization of anime sketches and speedup of colouring is faster than that of manual operation. In this they used a Swish-gated Residual block which is an improvement of the Residual block. They used the VGG network to extract the perceptual features.

Objective function(loss function) for this paper is: They use perceptual loss and the per-pixel loss are used to constitute the final loss of SGRU.

$$L_g = \sum_{l} \lambda_l ||\varphi_l(T) - \varphi_l(G)||_1$$

ϕl(T) is the VGG19 *lth* layer output of the original colour image and ϕl(C) is the VGG19 output of the generated image. λl is a hyper parameter. Discriminator loss is

$$L_d = -E[\log(\sigma(D(G)) + \log(1 - \sigma(D(T)))]$$

The experiment result of this paper proved SGRU can generate a more vivid and saturated color image than u-net and residual u-net.

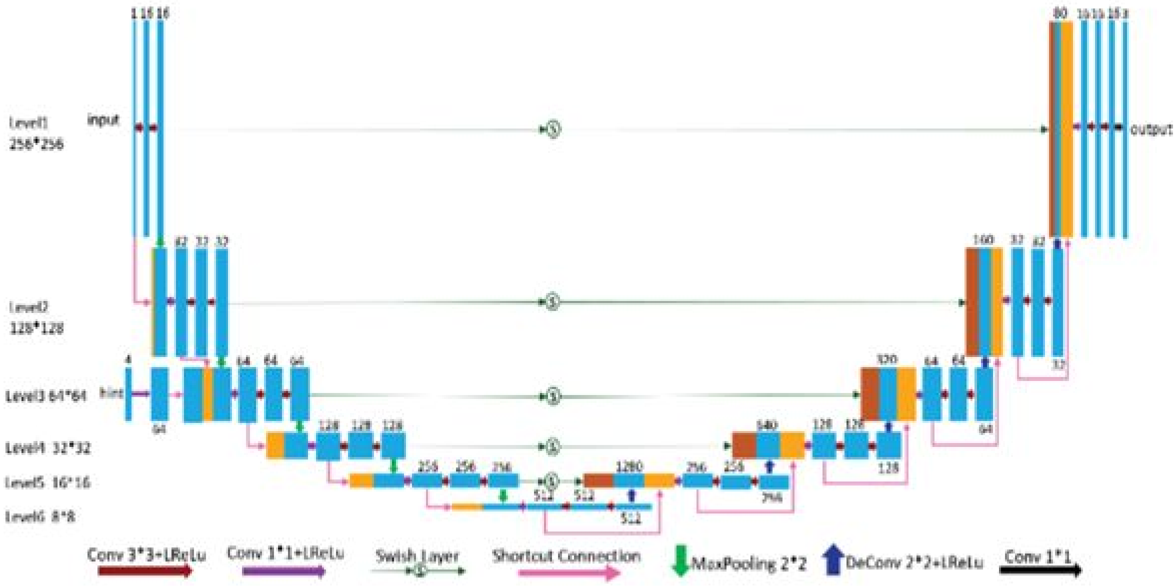
In Ru-Ting Ye[5] they proposed anime sketch colorization with hints. They proposed a colorization network to expect the network can generate a color image that is consistent and vivid with the color that the user wants. They propose the same network as mentioned in [4] but add hint color images in swish U-Net to give users wanted color. For hint images they use real color images to randomly generate color hints.

To create a color hint mask steps:

1. Take real image and 4 times down sampled new image size is (w/4)*(w/4) *3 (w image width of square image).

2. Create a random matrix of size (w/4) * (w/4) *1 the values between 0 and 1. When the values are greater than mask value then set value to the 1 otherwise 0.

3. Concatenate 1st step matrix and 2nd step matrix which size is (w/4)*(w/4)*4 add this matrix as the hint matrix in the U-Net.



These experiments show desired color by the user and can achieve good coloring results.

In[6] give the tips and tricks for the train GAN network. Normalizing input between [-1, 1] use tanh as the last layer of the generator.

In [7] tell us about how to train a GAN network using Keras. It tells us how to do keras code for training GAN network using simple MINIST handwritten digit reorganization data.

In [8] explain how to implement anime colorization using the GAN network.

I decided to go with [3] model. Where they used swish gated U -net and Spectrally Normalized GAN model used.

Performance Metric:

Pixel Difference based Methods

1. Mean Square Error- MSE is computed by averaging the squared intensity of the original (input) image and the resultant (output) image pixels

$$MSE = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m,n)^2$$

e(m, n) Error difference between original and generated image.

2. Peak signal-to noise Ratio: Mathematical measure of image quality based on pixel difference between two images

$$PSNR = 10log_{10}(\frac{(L-1)^2}{MSE}) = 20log_{10}(\frac{L-1}{RMSE})$$

3. Structure Similarity (SSIM) : Instead of doing pixel wise independent similarity in SSIM take a group of pixels and try to determine if two images are different or not. It is between -1 and 1. Positive 1 is perfect match and -1 for imperfect match.

References:

[1] Ian J. Goodfellow , Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair‡ , Aaron Courville, Yoshua Bengio, “Generative Adversarial Nets”, *2014.*

[2] Y. Liu, Z. Qin, T. Wan, and Z. Luo, “Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks,” Neurocomputing., vol. 311, no. 15 October 2018, pp. 78–87, 2018.

[3] L. Zhang, Y. Ji, X. Lin, “Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN,” Proceedings of Asian Conference on Pattern Recognition, 2017

[4] G. Liu, X. Chen, and Y. Hu, “Anime Sketch Coloring with SwishGated Residual U-Net,” Computational Intelligence and Intelligent Systems, p. 190-204, 2019.

[5] Ru-Ting Ye,Wei-Li Wang,Ju-Chin Chen and Kawuu W. Lin, “Interactive Anime Sketch Colorization with Style Consistency via a Deep Residual Neural Network” .

[6] Tip and tricks of GAN https://github.com/soumith/ganhacks

[7] Build GAN using Keras https://medium.com/datadriveninvestor/generative-adversarial-network-gan-using-keras-ce1c05cfdfd3 .

[8] Blog anime colorization https://towardsdatascience.com/sketch-to-color-anime-translation-using-generative-adversarial-networks-gans-8f4f69594aeb

[9] Unet Architecture https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/