



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569

Sketch to Color Image Using Generative Adversarial Networks

Dr. Niraj Upadhayaya¹, Ch.Vibhav Raju², Md. Junaid Subhan³, V. Shiva Teja⁴, V. Kushal Kumar⁵

Professor, Department of Computer Science and Engineering, JB Institute of Engineering & Technology,
Yenkally, Moinabad Mandal, R.R Dist-75, Telangana, India¹

B.Tech Student, Department of Computer Science and Engineering, JB Institute of Engineering & Technology,
Yenkally, Moinabad Mandal, R.R Dist-75, Telangana, India^{2,3,4,5}

ABSTRACT: Realistic image generation using deep neural networks has become a hot topic in machine learning and computer vision. Learning to generate colourful cartoon images from black-and-white sketches is not only an interesting research problem, but also a potential application in digital entertainment. Here investigate the sketch-to-image synthesis problem by using conditional generative adversarial networks (cGAN). We propose the auto-painter model which can automatically generate compatible colours for a sketch.

Existing approaches either need exact edge maps, or rely on retrieval of existing photographs. Our model accepts a 256x256 px black and white sketch image and predicts the coloured version of the image without knowing the ground truth. The model will be trained on the Anime Sketch-Colorization Pair Dataset available on Kaggle which contains 14.2k pairs of Sketch-Colour Anime Images.

Conditional adversarial networks not only learn the mapping from input image to output image, but also learn a loss function to train this mapping. This makes it possible to apply the same generic approach to problems that traditionally would require very different loss formulations. We demonstrate that this approach is effective at synthesizing and coloring images.

KEYWORDS: Machine Learning, General Adversarial Networks, Cartoon images, Gradient Descent, Generator, Discriminator.

I. INTRODUCTION

Human beings possess a great cognitive capability to comprehend black-and white cartoon sketches. Our mind can create realistic colourful images when we see black-and-white images. Sketching is the major hobby of teenagers. People usually draw characters of their favourite cartoons in their free time. It needs a great artistic talent to choose appropriate colours, also with proper changes in light and shade to create vivid cartoon images. In the current times, sketches are majorly influenced by Japanese Anime characters. Anime are the Japanese style of story telling through books which contain instances of story realm. These anime characters are given life with good colour combinations and shadowing. Through colouring we give a complete look to a sketch. Sketch is a skeleton, colouring it puts life into sketch. In real life we use pastels, water colours, colour pencils, sketches, oil paints, etc to colour our subject.

How to automatically paint the sketch into colourful cartoon image is a useful application for digital entertainment. In this work, we are interested in solving this problem with the help of deep neural networks. Here, we try to colour a sketch artificially without a creative human being! In particular, we use GAN and try to colour new images by training the model on coloured images.

II. RELATED WORK

Generative adversarial networks (GANs) were recently regarded as a breakthrough in machine learning, it consists of two 'adversarial' modules: a generative module G that captures the data distribution, and a discriminative module D that estimates the probability that a sample came from the training data rather than G. Both G and D could be

deepneural networks. In image synthesis with GANs, the generator attempts to produce a realistic image from a random input image to fool the simultaneously adversarial trained discriminator, which tries to distinguish whether its input image is from the training set or the generated set. The generator has benefited from convolutional decoder networks, it can go back to the work using deep convolutional decoder networks to generate realistic images.

GANs are unconditioned generative models that learn a mapping from random noise vector z to output y : $G : z \rightarrow y$. In contrast, conditional GANs (cGANs) learn a mapping from observed input x and random noise vector z , to y : $G : \{x, z\} \rightarrow y$. Several works studied different cGAN where the generator is conditioned on inputs such as text, labels and other forms of the images to generate 'fake' images. The discriminator needs to distinguish the authentic and fake pairs of images. In this paper, both of generator and discriminator are conditioned on the input sketch in order to get a better supervised performance.

The 'pix2pix' method proposed in is a 'U-net' architecture to deal with general image-to-image transfer which allows the decoder to be conditioned on encoder layers to get more information. They investigated different kinds of image-to-image transfer tasks that include transforming the image of daylight to night, produce city images from map, and even synthesize shoes and handbags from designer's sketches. Sangkloy proposed the 'scribbler' by allowing users to scribble over the sketch to indicate preferred colour for objects.

III. METHODOLOGY

The GAN model architecture is made up of two sub-models: a generator model for creating new instances and a discriminator model for determining whether the generated examples are real, from the domain, or fake, generated by the generator model.

- **Generator.** Model that is used to generate new plausible examples from the problem domain.
- **Discriminator.** Model that is used to classify examples as real or fake.

The Generator Model

The generator model generates a sample in the domain using a fixed-length random vector as input. A Gaussian distribution is employed to generate the vector, which is then used to seed the generative process. After training, points in this multidimensional vector space will correspond to points in the problem domain, forming a compressed representation of the data distribution. This vector space is referred to as a latent space, or a vector space comprised of latent variables. Latent variables, are those variables that are important for a domain but are not directly observable.

The Discriminator Model

The discriminator model takes an example from the domain as input and predicts a binary class label as real or fake image. The real example comes from the training dataset. The generated examples are output by the generator model. The discriminator is a normal classification model.

After the training process, the discriminator model is discarded as we are interested in the generator. Sometimes, the generator can be repurposed as it has learned to effectively extract features from examples in the problem domain. Some or all of the feature extraction layers can be used in transfer learning applications using the same or similar input data.

Structured losses for image modelling

Image-to-image translation problems are often formulated as per-pixel classification. These formulations treat the output space as "unstructured" in the sense that each output pixel is considered conditionally independent from all others given the input image. Conditional GANs instead learn a structured losses which penalize the joint configuration of the output. Many books has considered losses of this kind, with methods including conditional random fields, the similarity between two images, feature matching, the convolutional pseudo-prior, nonparametric losses, and losses based on matching covariance statistics. The conditional GAN is different in that the loss is learned, and can, in theory, penalize any possible structure that differs between output and target.



Fig. 3.1 Different losses induce different quality of results.

Objective

The objective of a conditional GAN can be expressed as

$$L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z)))] \rightarrow (1)$$

Here ‘z’ is Gaussian noise.

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. $G^* = \arg \min_G \max_D L_{cGAN}(G, D)$.

To test the importance of conditioning the discriminator, we also compare to an unconditional variant in which the discriminator does not observe x:

$$L_{GAN}(G, D) = E_y[\log D(y)] + E_{x,z}[\log(1 - D(G(x, z)))] \rightarrow (2)$$

Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss, such as L2 distance. The discriminator’s job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground truth output in an L2 sense. We also explore this option, using L1 distance rather than L2 as L1 encourages less blurring:

$$LL1(G) = E_{x,y,z}[|y - G(x, z)|] \rightarrow (3)$$

Our final objective is

$$G^* = \arg \min_G \max_D L_{cGAN}(G, D) + \lambda LL1(G) \rightarrow (4)$$

Existing System

- The existing system is a sketch in electronic format, which is not colored. We can color the sketch using editing tools like Microsoft Paint tool ,Photoshop ,Gimp and other Image editing software.
- Though we can colour manually ,our task is to automate coloring process and develop a model that can intelligently colour the given sketch with pleasing colours and colour each section naturally, that would give a look of manual colouring.

Proposed System

- The proposed system is a model that can color the sketch realistically using a set of colors that would ideally suit the parts of sketch.
- This coloring process is done by the model itself ,without any external help. We can achieve this using GAN, short for Generative Adversarial Networks.

Applications

This System has higher significance in Digital Media.Manga which is generally published in greyscale, has frames which produce results requiring less colour correction by artist since more information is present in the input compared to line art.Hence, greyscale requires less effort to automatically colorize frames in each page.For minimal extra cost, studios can publish coloured versions of their popular manga.

System Architecture

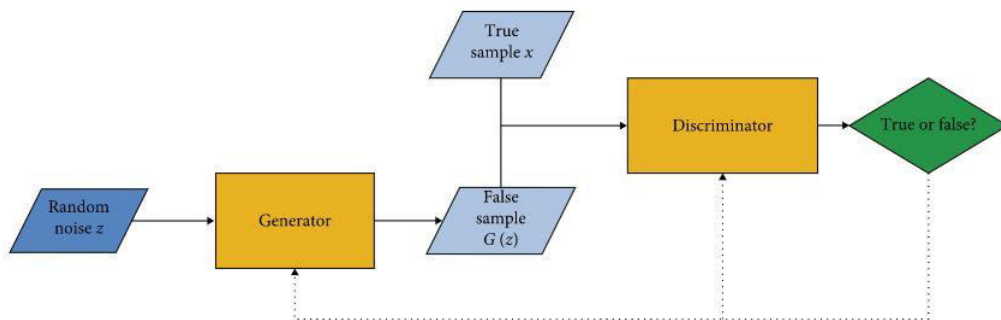


Fig. 3.2 System Architecture

System Modules

1. Data Collection:

The dataset contains coloured sketch images which are to be preprocessed. The dataset contains 14.2k pairs of sketch-colour anime images taken from Kaggle website.

2. Data Pre-processing:

Data preprocessing is done to ensure the uniformity of image size, if by change there is a different sized image in dataset. It is done in 4 stages.

- resize(): This function converts different sized images into 286x286 px from 512x512 px.
- random_crop(): This function returns the cropped input and real images which have the desired size of 256x256 px.
- normalize(): This function, as the name suggests, normalizes images to [-1, 1].
- flip(): flips random images left or right to increase the contrast between images.

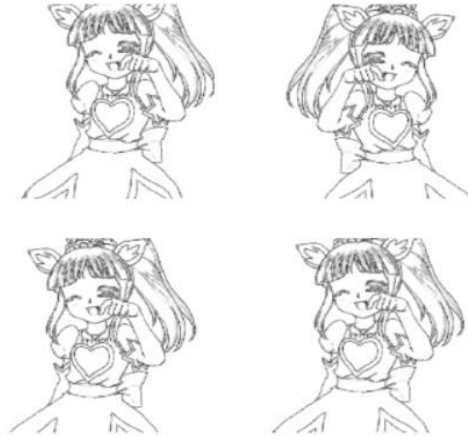


Fig. 3.3Pre-processed image

3. Building Generator Model:

At this stage we build our Generator model which is basic part of GAN. It takes black and white sketch images of 256x256 px and outputs the similar coloured image to the input.

4. Building Discriminator Model:

The discriminator basically checks if the input image is actually from the input dataset or from the generator. The generator makes it hard for the discriminator to distinguish real image from fake image.

5. Optimization:

The process of optimization requires algorithms or methods that change the attributes of our neural network such as weights and learning rates in order to reduce losses. Adam Optimizer is the best one, which is able to optimize every use case.

6. Fitting the model: Here we fit the model using previously created generator and discriminator sub models. We make connections between generator and discriminator in this step, according to the proposed architecture. Finally, our model is trained and ready to be tested.

7. Testing Outputs:

This randomly selects 5 images from the test_dataset and inputs them individually to the Generator Model. Now the model is well trained and predicts nearly perfectly coloured images of the input sketch images.

IV. EXPERIMENTAL RESULTS

Figures shows the results of Sketch to Colour image conversion by using Generative Adversarial Networks. Figure (A), (B), (C) in each section represent Input mage, Ground Truth and Predicted Image respectively.



Fig. 1. Sketch to Coloured Image Conversion (a) Input Image (b) Ground Truth (c) Predicted Image after giving input image to the model

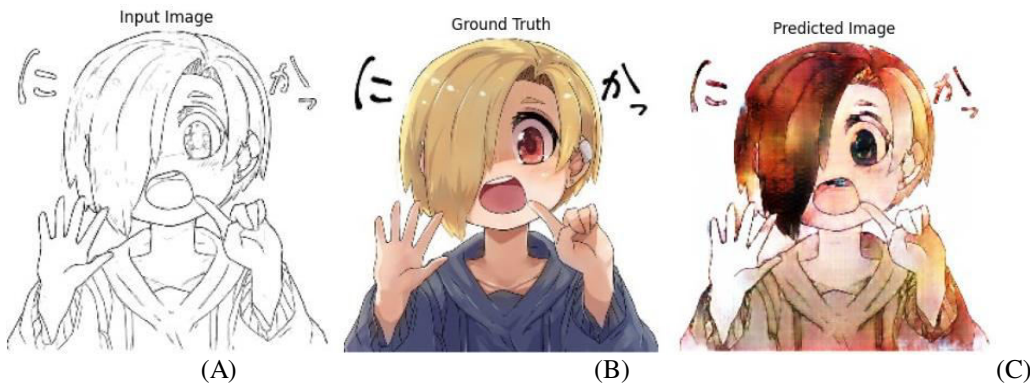


Fig. 2. Sketch to Coloured Image Conversion (a) Input Image (b) Ground Truth (c) Predicted Image after giving input image to the model

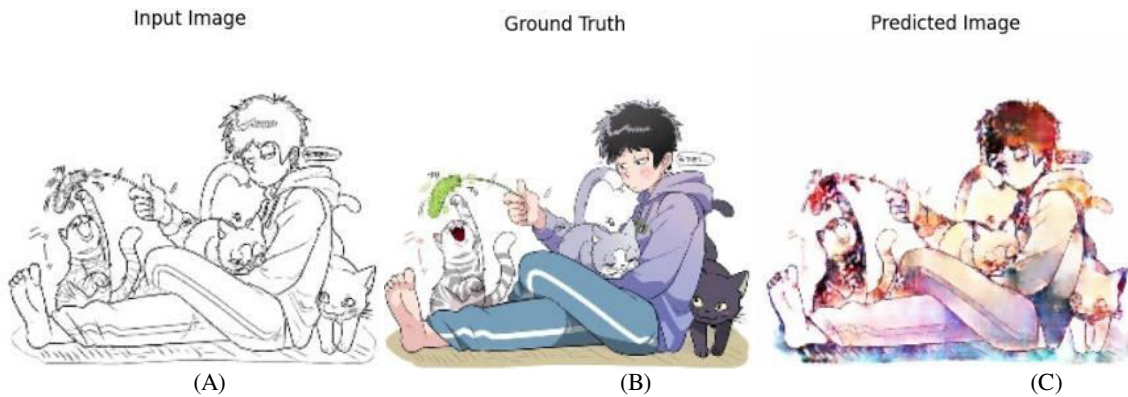


Fig. 3. Sketch to Coloured Image Conversion (a) Input Image (b) Ground Truth (c) Predicted Image after giving input image to the model



V. CONCLUSION

The task of mapping sparse inputs to coloured outputs using learning-based approaches is difficult. We have successfully built a Conditional Generative Adversarial Network model which accepts a black and white sketch image and predicts the coloured version of the image. If artists make detailed sketches, they can bypass the colorization process of crucial frames and in between frames thereby speeding up the production process while reducing the overall cost. Rather than employing colouring artists to work on sketches we can automate this process using our model, thereby reducing man-power significantly and also reduce cost. The incorrectly coloured parts of the sketch can be later reviewed and re-coloured by artists.

REFERENCES

- [1] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Nets (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.
- [2] Sajjadi, Mehdi S. M.; Schölkopf, Bernhard; Hirsch, Michael (December 23, 2016). "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis". arXiv:1612.07919. Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, Stanley Osher, "Simultaneous Structure and Texture Image Inpainting", IEEE Transactions On Image Processing, vol. 12, No. 8, 2003.
- [3] Generative Adversarial Networks, an article on medium <https://towardsdatascience.com/generative-adversarial-networks-gans-89ef35a60b69>
- [4] Research Paper on Image-to-Image Translation with Conditional Adversarial Networks by Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros <https://arxiv.org/abs/1611.07004>
- [5] A Research Paper on Generative Adversarial Networks by Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio <https://arxiv.org/abs/1406.2661>
- [6] Tang, Xiaou; Qiao, Yu; Loy, Chen Change; Dong, Chao; Liu, Yihao; Gu, Jinjin; Wu, Shixiang; Yu, Ke; Wang, Xintao (September 1, 2018). "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks". arXiv:1809.00219. Bibcode:2018arXiv180900219W.



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details