MangaCa Literature Review

09-09
Chen Zerui (4S2) & WangYiqin (4S1)

Due to the lack of grayscale values in manga line arts, manga line art colourisation has been a challenging computer vision problem. The generalisation of machine learning models to this type of computer vision problem has also been made difficult due to the lack of authentic (human drawn) line arts and coloured line art pairs [2]. In order to tackle this complex machine learning problem, researchers have developed the GAN (Generative Adversarial Networks) architecture. For instance, the readily available and popular PaintsChainer network developed by [4] is able to achieve stunning results for both automatic and semi-automatic (user-guided) colourisation of manga line arts. [4] Achieved such a result through the usage of U-Net as its generator network, which consists of an encoder-decoder architecture that includes skip-connections across the convolutional blocks between the encoder and decoder.

Later works like the one from [6] are based upon the basic U-Net structure of the PaintsChainer networks but included multiple auxiliary networks to further expand the capacity of the colourisation network and achieve more realistic results. In [6], auxiliary networks utilised includes the Illust2Vect network from [3], a VGG-16 based convolutional network trained to tag anime images with over 1500 unique labels. The Illust2Vect network functions as a features network which helps identify key features of the manga line arts and their corresponding coloured pairs in order to compute a content loss that has shown to improve the network's ability to retain semantic information from the original sketch images. At the same time, outputs of the features network are also included in the inputs to the generator and discriminator network to improve training.

In other works such as [7], semi-automatic colourisation of line arts is enabled through accepting an additional 'hint' image, which is preprocessed by a VGG-19 network pre-trained on the ImageNet dataset and subsequently concatenated into the connecting layers between the encoder and decoder. The 'hint' image used can either be a separate fully coloured manga character image or a 4-channel input of coloured dots. [7] has also identified a potential vanishing-gradient problem caused by the skip connections found in a typical U-Net structure.

According to [7], due to the need to preserve a large portion of the information made available by the original line art image, U-Nets tend to simply deactivate higher level convolutional layers, resulting in no gradient being trained in the middle layers of the generator network. [7] overcame this problem through the use of additional auxiliary 'guided-decoder', which compute additional MSE (mean square error) losses that are propagated through the middle convolutional layers to avert the vanishing-gradient problem.

As of late, a new type of GAN, ProGAN, has been developed by Nvidia [1]. In their research, [1] applied the basic generator-discriminator of a GAN network but instead of training all layers of the network at once, the generator and discriminator, as mirror images of each other, always grow in synchrony. This has allowed the GAN to generate images at higher resolutions as the entire network grows. According to [1], the progressive growing of GAN has substantially increased the stability of the training due to less class information and fewer modes. The gradual increase of resolution has also significantly increased the stability of training, allow megapixel-scale (high resolution) images to be successfully generated.

Due to the development of ProGAN, which has not yet been applied to colourisation tasks, this project aims to investigate the competency of the new ProGAN architecture at manga line art colourisation while applying additional auxiliary networks used in older GAN related paper that has shown to improve the capacity of the network. The network can be trained on the Danbooru 2018 dataset from [5] which provides over 250k authentic coloured anime drawings.

References

1. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION. Doi:1710.10196
2. Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. (2018, August 10). *User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks* [Scholarly project]. In *User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks*. Retrieved June 14, 2019.

3. Saito, M., & Matsui, Y. (2015). *Illustration2Vec: A semantic vector representation of illustrations.*[Scholarly project]. Retrieved June 6, 2019.
4. Yonetsuji, T. *Paintschainer*. Retrieved June 6, 2019, from github.com/pfnet/PaintsChainer
5. Branwen, G., & Gokaslan, A. (2019, January 2). *Danbooru2018: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset.* Retrieved May 25th, 2019.
6. Lvmin, Z., Yi, J., & Xin, L. (2017, June 13). *Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN.* Retrieved April 27th, 2019.
7. Lvmin, Z., Chengze, L., Tien-Tsin W., Yi, J., & ChunPing, Liu.. (2018, November). *Style Two-stage Sketch Colorization.* Retrieved May 20th, 2019.

MangaCa

Grp 09-09

Investigating the use of Progressive Generative Adversarial Networks for the colourisation of manga images

Chen Zerui (4S2) Wang Yiqin (4S1)

Abstract

Due to the lack of grayscale values in manga line arts, manga line art colourisation has been a challenging computer vision problem. The generalisation of machine learning models to this type of computer vision problem has also been made difficult due to the lack of authentic (human drawn) line arts and coloured line art pairs. To achieve good results in image colourisation tasks, researchers have successfully applied the GAN (Generative Adversarial Networks) architecture. Recently, a new modified version of the GAN architecture, ProGAN, published by [1], has achieved astonishing results in producing realistic automatically generated images of human faces. In their research, [1] claims that the new architecture and training method improved training stability and allows for easy upscaling of output image resolutions. As such, this project aims to investigate the competency of the new ProGAN architecture at manga line art colourisation. We procured a large sample size (approximated 250k) of authentic coloured manga. Our proposed model was able to converge significantly faster than other existing models and we achieved realistic colourisation of manga line arts, comparable to the widely available PaintsChainer network from [5].

1. Introduction

Our project revolves around 2 premises.

Firstly, we aim to simplify and streamline the process of line art colourisation. Manga line art colourisation is key to the production of artistic works such as illustrations and animation. Despite advancements in computer image editing solutions such as PhotoShop, the work of colouring line arts still needs to be performed manually in a painstaking fashion. Although many recent projects have produced astonishing results in terms of precision and realism of manga colourisation [3, 5, 8]. Many of them are purely research-based and are not made available to artists. While [5] does provide a robust online colourisation platform, the results produced by [5] is not nearly realistic enough for use in actual artwork or animation productions.

Secondly, we aim to investigate the proficiency of the new ProGAN network developed by [1]. In their research conclusion, ProGAN was said to be able to produce highly realistic images of human faces, comparable to existing GAN based models but at a much higher resolution (1024x1024). At the same time, [1] reveals that the ProGAN network is able to achieve convergence much quicker and is more stable during training time. As the ProGAN network is a relatively new architecture, this project aims to investigate its proficiency at colourisation tasks, which is fundamentally different from image generation tasks.

In this project, we extended from the foundations of the ProGAN architecture of [1]. Unlike in [1], where the network accepts noise image as input, our network accepts a full grayscale line art as input. As a secondary input, the network accepts a fully colourised image as ‘hint’. Features of the ‘hint’ image are extracted using the network from [3] and guide the network in its colourisation. The network can also function without a ‘hint’ input. Auxiliary networks like those found in [8] are also included for the purpose of providing extra features and semantic information to the network. In between convolutional layers found within the generator, ResNeXt blocks were utilised to extend the capacity of the network. Additional features extractors are also used to compare generated images with the authentic coloured images, generating a content loss to help the network retain important features found in the line art. A

perceptual loss is generated by the discriminator, which consists of an encoder structure using multiple convolutional layers, producing a single probability value as an output.

We trained our network with 250k coloured images procured from [7]. For each coloured image, a corresponding line art image is generated with the use of [6]. Authentic line art to coloured image pairs is also included in the dataset.

We also compared the results of our network with the popular PaintsChainer network from [5] by comparing 1000 authentic colourised images and 1000 generated illustrations from our network using Frechet Inception Distance [10].

2. Related Works

Due to the lack of grayscale values in manga line arts, manga line art colourisation has been a challenging computer vision problem. The generalisation of machine learning models to this type of computer vision problem has also been made difficult due to the lack of authentic (human drawn) line arts and coloured line art pairs [3]. In order to tackle this complex machine learning problem, researchers have developed the GAN (Generative Adversarial Networks) architecture. For instance, the readily available and popular PaintsChainer network developed by [4] is able to achieve stunning results for both automatic and semi-automatic (user-guided) colourisation of manga line arts. [4] Achieved such a result through the usage of U-Net as its generator network, which consists of an encoder-decoder architecture that includes skip-connections across the convolutional blocks between the encoder and decoder.

Later works like the one from [5] are based upon the basic U-Net structure of the PaintsChainer networks but included multiple auxiliary networks to further expand the capacity of the colourisation network and achieve more realistic results. In [5], auxiliary networks utilised includes the Illust2Vect network, a VGG-16 based convolutional network trained to tag anime images with over 1500 unique labels. The Illust2Vect network functions as a features network which helps identify key features of the manga line arts and their corresponding coloured pairs in order to compute a content loss that has shown to improve the network's ability to retain semantic information from the original sketch images. At the same time, outputs of the features

network are also included in the inputs to the generator and discriminator network to improve training.

In other works such as [8], semi-automatic colourisation of line art is enabled through accepting an additional ‘hint’ image, which is preprocessed by a VGG-19 network pre trained on the ImageNet dataset and subsequently concatenated into the connecting layers between the encoder and decoder. The ‘hint’ image used can either be a separate fully coloured manga character image or a 4-channel input of coloured dots. [8] has also identified a potential vanishing-gradient problem caused by the skip connections found in a typical U-Net structure. According to [8], due to the need to preserve a large portion of the information made available by the original line art image, U-Nets tend to simply deactivate higher level convolutional layers, resulting in no gradient being trained in the middle layers of the generator network. [8] overcame this problem through the use of additional auxiliary ‘guided-decoder’, which compute additional MSE (mean square error) losses that are propagated through the middle convolutional layers to avert the vanishing-gradient problem.

As of late, a new type of GAN, ProGAN, has been developed by Nvidia [1]. In their research, [1] applied the basic generator-discriminator of a GAN network but instead of training all layers of the network at once, the generator and discriminator, as mirror images of each other, always grow in synchrony. This has allowed the GAN to generate images that are subsequently higher resolution as the entire network grows. According to [1], the progressive growing of GAN has substantially increased the stability of the training due to less class information and fewer modes. The gradual increase of resolution has also significantly increased the stability of training, allow megapixel-scale (high resolution) images to be successfully generated.

3. Proposed Method

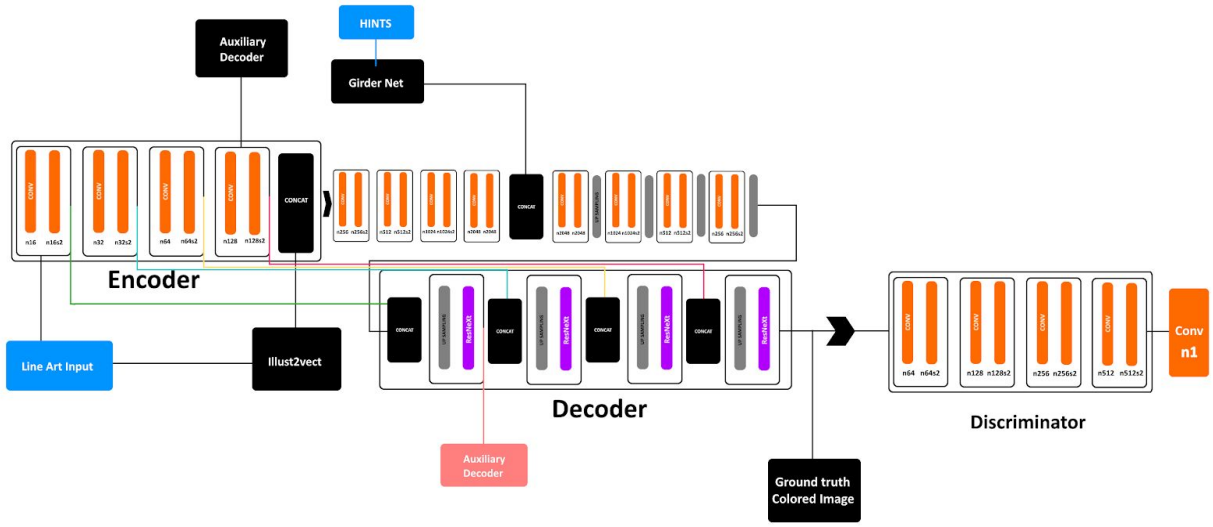

Fig. 1
Visual representation of our ProGAN network

In this project, the generator adopts the architecture of an enhanced residual U-net. The generator accepts 2 inputs, a grayscale line-art image and a hint image. A discriminator matching the architecture of the encoder is used to tell whether its inputs are fake images from the generator or real images drawn by artists.

3.1 Generator Architecture

The detailed structure of our network is shown in fig. 1.

The generator is based upon an enhanced residual U-net utilising skip-connections between convolutional layers to help retain special and semantic information of the outputs of previous layers. Two additional auxiliary decoders are added to the generator network (before and after the connecting layers) as proposed by [8] to avoid vanishing gradients. The auxiliary decoder before the connecting layers upscales the output of the convolutional layer to a dimension of 1×512×512 (grayscale) while the auxiliary decoder attached after the connecting layer provides an output of shape 3×512×512 (rgb). A content loss is obtained during training from both of these networks by calculating the MSE (mean square error) between the outputs and the ground truth sketch images and coloured images. The respective losses are then propagated through the previous layers.

The generator also accepts a hint input which is pushed through a feature-extractor network. The feature-extractor network is based on a series of convolutional layers that eventually provides an output tensor of shape 4×4×2048 which is subsequently fed into the connecting layers of the generator network. The architecture of the feature extractor network is adapted from the girder net used in [9].

To maximise the capacity of our network, ResNeXt blocks are used in the generators along with the convolutional networks. Cardinality of the RestNeXt blocks is set to a constant of 8.

3.2 Discriminator Architecture

The architecture of the discriminator is near identical to the decoder section of the generator to enable easy progressive growing of the GAN.

In the discriminator, a series of convolutional layers downscales the input (output from the generator or real coloured image) of dimensions 3×512×512 to a single output value with softmax activation to limit the output value between 0 and 1.

3.2 Loss Functions

For the generator, the complete loss is defined as:

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

Where the x, y is the paired domain of sketches and paintings, and V(x) is the output of the feature extractor's fully connected layers without Relu activation, and G_f (x, V (x)) is the final output of the generator, the G_{g1}(x) and G_{g2}(x, V (x)) are the outputs of the two guide decoders at the entry and the exit of mid-level layers accordingly. From [8], the recommended value of α and β is 0.3 and 0.9.

The complete loss of the GAN network, taking into account the output of the discriminator, is given by:

$$L_{GAN}(V, G_f, D) = \mathbb{E}_{y \sim P_{data}(y)}[Log(D(y))] + \mathbb{E}_{x \sim P_{data}(x)}[Log(1 - D(G_f(x, V(x))))]$$

Where D is the discriminator.

The objective of the network can be represented by:

$$G^* = \arg \min_{G_f} \max_{D} L_{GAN}(V, G_f, D) + \lambda L_{l1}(V, G_{f,g_1,g_2})$$

4. Experimentation

As per the specification of the ProGAN architecture, our network was trained incrementally, growing from depth of 1 to 4, where depth determines the number of convolutional blocks in both the encoder and decoder of the generator network. The growth is incremented by 1 at the following steps [5k, 10k, 50k, 100k]. The learning rate for all networks is 2×10^{-4}. In total, the networks are trained on a total of 250k images from [7], where for each coloured image, a corresponding line art image is generated with the use of [6].

A diagram showing the loss curves during the training is shown in fig. 2. The system of networks approached convergence around the 150k step.

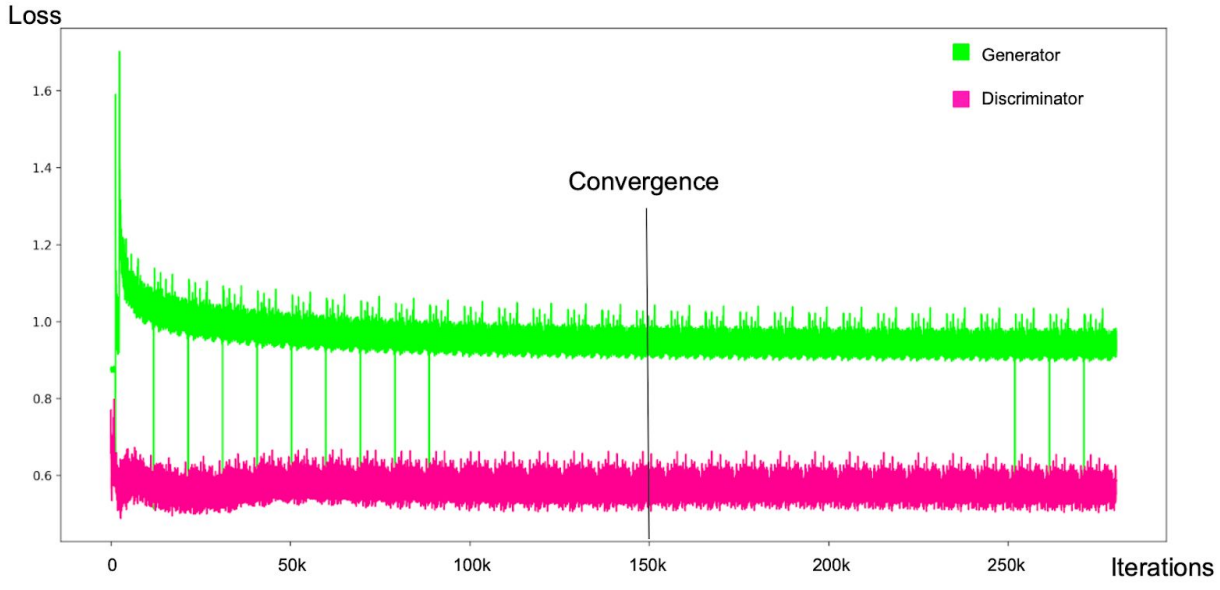

Fig. 2

Loss curves of generator (green) and discriminator (magenta) over number of training steps.

5. Results and discussion

In order to evaluate our network's performance against the readily available and popular online colorization tool PaintsChainer [5], the Fréchet Inception distance [10] was used as the metric of performance as shown in fig. 3. We first evaluated the Fréchet Inception distance between 1000 images generated by our network without using hints and their corresponding original colored images, chosen randomly from the Danbooru2018 dataset. A closer Fréchet Inception distance indicates that the two sets of images are closer perceptually - the objective colorization networks aim to achieve. Subsequently, we evaluated the distance between 100 images generated without hint using the "Canna" model of PaintsChainer and their respective original colored images. The dataset chosen for evaluating PaintsChainer's Canna model is much smaller, as PaintsChainer only offers an online GUI which requires manual operation to generate the colorized images. Nonetheless, the Fréchet Inception distance showed a definitive increase in performance of our network compared to PaintsChainer's Canna, with each network scoring a distance of 95.3 and 221.1 respectively.

We have also included samples of sketches colorized by our network - with and without hint in fig. 4 and fig. 5 respectively.

Model	FID
PaintsChainer (canna)	181.07
PaintsChainer (tanpopo)	130.76
PaintsChainer (satsuki)	123.82
Ours (ProGAN)	95.26

Fig. 3
Table showing the respective Fréchet Inception distance of various models

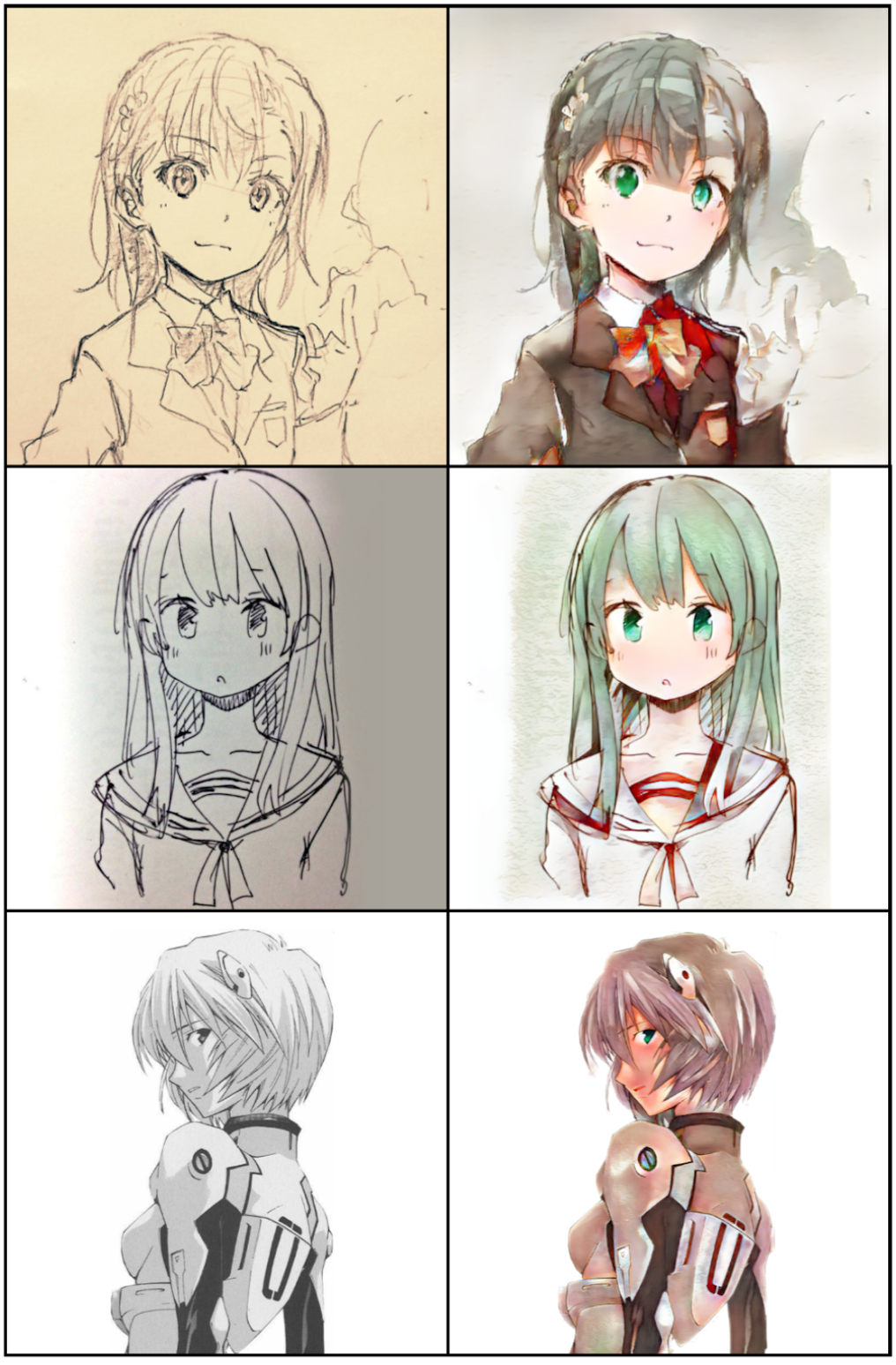

Fig. 4

Sample of images colourised without hint

Fig. 5
Sample of images colourised with hint

6. Conclusion

In conclusion, this project explores the adaptability of the ProGAN architecture at performing the task of anime-styled sketch colorization, where it has managed to obtain impressive results rivaling certain state-of-the-art solutions for both hinted and fully-automatic colorization.

7. Reflection

Through this project, our team has not only refined our technical knowledge, but also learnt more about effective teamwork, as well as the benefits of planning ahead of time. As the data collection, training and fine-tuning of deep machine learning models, especially GANs, are all highly time consuming, it was critical for us to effectively carry out these tasks to meet deadlines.

References

1. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION. Doi:1710.10196
2. Wolf, S. (2018, December 16). ProGAN: How NVIDIA Generated Images of Unprecedented Quality. Retrieved March 3, 2019, from https://towardsdatascience.com/progan-how-nvidia-generated-images-of-unprecedented-quality-51c98ec2cbd2
3. Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. (2018, August 10). *User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks* [Scholarly project]. In *User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks*. Retrieved June 14, 2019.
4. Saito, M., & Matsui, Y. (2015). *Illustration2Vec: A semantic vector representation of illustrations.*[Scholarly project]. Retrieved June 6, 2019.
5. Yonetsuji, T. *Paintschainer*. Retrieved June 6, 2019, from github.com/pfnet/PaintsChainer
6. Illyasviel, *sketchKeras*. Retrieved June 5, 2019, from https://github.com/lllyasviel/sketchKeras
7. Branwen, G., & Gokaslan, A. (2019, January 2). *Danbooru2018: A Large-Scale Crowdsourced and Tagged Anime Illustration Dataset.* Retrieved May 25th, 2019.
8. Lvmin, Z., Yi, J., & Xin, L. (2017, June 13). *Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN.* Retrieved April 27th, 2019.
9. Lvmin, Z., Chengze, L., Tien-Tsin W., Yi, J., & ChunPing, Liu.. (2018, November). *Style Two-stage Sketch Colorization.* Retrieved May 20th, 2019.
10. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems* (pp. 6626–6637).