Psychedelic Art Generation Using CycleGAN

Sthithpragya Gupta
Universitat Jaume I
Castellon de la Plana, Spain
sthithpragyagupta@gmail.com

Abstract

The objective of this study was to generate psychedelic art about landscapes. The art-style was to be learnt from the psychedelic images found over the internet. To achieve this, a Cycle-Consistent Adversarial Network (CycleGAN) was developed and implemented. To train this framework, data-sets of images were prepared before hand. Different amounts of training of the network translated to different types of learnt art-styles. These learnt styles have then been used to generate art about landscapes by taking as input an original landscape image and outputting their artistic rendition.

1 Introduction

The presence of an algorithmic basis for art generation in the human psyche remains a topic for debate. However, the fascination with algorithm driven processes for generating novel art has been steadily increasing. Consequently, in the recent years, machine learning algorithms are finding new applications in the domain of art and design. One such technique of art generation is *art-style transfer*. The objective of this technique is to learn the art-style of an image and render or recreate another image in that style. The process is akin to commissioning an artist to make a rendition of an image of your choice in his/her own style. In this study, the objective was to carry out the process of art-style transfer to learn the art-style of popular psychedelic art found on the internet and then recreate actual photographs of landscape, and natural scenery in that style. This section discusses the conventional technique of art-style transfer and then introduces the Cycle Generative Adversarial Network (CycleGAN) which has been employed in this study to achieve the aforementioned objective.

1.1 Conventional art-style transfer

The *Neural Algorithm of Artistic Style* presented in [Gatys *et al.*, 2015] is an optimization technique to achieve art-style transfer. The framework utilises a Deep Convolutional Neural Network (DeepCNN) which takes as input a user specified pair of images. One of these images is the content image and the other is a style reference image (such as an artwork by a famous painter). The framework begins with an empty base image. It defines two loss functions - *Lcontent* and *Lstyle*. Lcontent measures the lack of similarity between the content of the base and content images whereas Lstyle measures the lack of similarity between styles of the base and style images. With each step, the DeepCNN transforms the base image by minimizing the content and style distances (losses) via back-propagation. This eventually transforms the base image into an image that matches the content of the content image and the style of the style image.

While capable of generating visually appealing art, the shortcoming of this architecture is that it requires the user to input a pair of images to develop the art. It can only transfer the art-style of one image onto another. A more comprehensive approach is to work with a data-set of unpaired content-style images and learn the art-style from a multitude of artworks so that this learned style maybe applied to any standalone content image.

1.2 Generative Adversarial Networks (GANs)

The two key approaches of statistical classification are the Generative and Discriminative approaches. In the Generative approach, the objective is to learn the joint probability distribution p(x,y) of the input x and label y. From the learned statistical model, the conditional probability p(x|y) can be computed which can be used for classification tasks (Generative Classification). The learned model can also be employed to generate random variables belonging to the aforementioned probability distribution (Generative Models). In contrast, the Discriminative approach directly models the conditional probability distribution p(x|y) to carry out the classification tasks (Discriminative Classification) [Ng and Jordan, 2002].

The Generative Adversarial Network (GAN) is a framework to develop a Generative Model via an adversarial process [Goodfellow *et al.*, 2014]. The architecture consists of two neural networks - Generator and Discriminator - which compete in a zero-sum game and train simultaneously until reaching the Nash equilibrium. The Generator assumes a joint probability distribution for the data and generates samples. The Discriminator classifies whether the sample belongs to the original training data or the generated data. During the training phase, the Generator trains to generate data as close to the training set as possible and maximise the probability of Discriminator wrongly classifying the sample data. On the contrary, the Discriminator trains to classify the sample data as accurately as possible. The training of these neural networks continues until the Discriminator can no longer discriminate between generated and training data. Once complete, the Generator has learned the stochastic model of the training data.

1.3 CycleGAN - Using GAN for art-style transfer

Cycle-Consistent Adversarial Network, also known as CycleGAN, is an extension of GANs which consists of four neural networks (two Generator-Discriminator pairs). The goal of the network is to learn a mapping ($G : X \rightarrow Y$) from source (X) to target (Y) images such that the distribution of images generated from G(X) is indistinguishable from the distribution Y, using an adversarial loss. Since this mapping is highly under-constrained, it is coupled with an inverse mapping ($F : Y \rightarrow X$) accompanied with a cycle consistency loss to enforce $F(G(X)) \approx X$ (and vice versa) [Zhu *et al.*, 2017]. The architecture of the network is explained in context to the objective task (translating landscape pictures to psychedelic images) in section 2.2.

2 Methodology and Implementation

The process of achieving the task objective was carried out in multiple steps. First, the training data-set consisting of images of psychedelic art and landscapes was compiled. Next, the architecture of CycleGAN was generated. Finally, the network was trained on the training data-set and results documented.

2.1 Preparing the training data-set

To prepare the image data-set for training the CycleGAN, the following steps were followed:

(i) **Sourcing images:** To prepare the initial data-set, the Python package *google_images_download* was used to perform a Google search and download the first 400 images associated with the keywords ”landscape wallpaper 1920x1080”. These images constituted the landscape image-set. Similarly, the psychedelic image-set was prepared using the keywords ”psychedelic art wallpaper 1920x1080”. This choice of keywords was made to have majority of the images with the same resolution and quality.

(ii) **Cleaning the image-sets:** From the 400 downloaded images per set, the following images were removed:

- Images in formats other than .jpeg
- Images with sizes other than 1920 pixels(px) by 1080px
- Images with watermarks
- Images with thick borders
- Artistic conceptions and fantasy images (for landscape images)
- Images with subject other than nature (for landscape images)
- Images of patterns and optical illusions (for psychedelic art)
- Images with excessive graffiti or text (for psychedelic art)

(iii) **Formatting to the correct size:** Post cleaning, 150 images were left in each of the image-sets. Images in each of the sets were then scaled down to 456px by 256px resolution from the original 1920px by 1080px resolution, while maintaining the aspect ratio (16:9). Next, these images were cropped to 256px by 256px size.

Ultimately we have two image sets - the *Landscape image data-set* (fig. 4) and the *Psychedelic art data-set* (fig. 5) with 150 images each. While preparing the data-set, images of resolution 1920px by 1080px were initially downloaded as this resolution offered the widest choice of landscape images and psychedelic art on the internet. Next, the images were scaled down and changed to a square aspect ratio to reduce the computational cost associated with training the CycleGAN on the images.

2.2 Developing the architecture for CycleGAN

As previously mentioned, the CycleGAN employs four neural networks viz –

(i) *Landscape Generator* LG to generate pictures of landscapes

(ii) *Psychedelic Generator* PG to generate psychedelic art

(iii) *Landscape Discriminator* LD to differentiate between generated (or fake) and real landscape pictures

(iv) *Psychedelic Discriminator* PD to differentiate between generated (or fake) and real psychedelic art

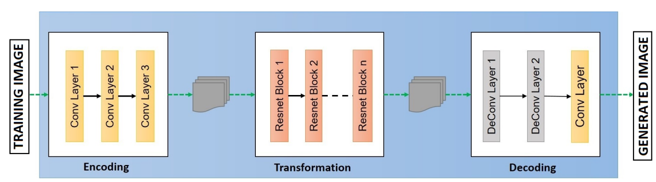

Figure 1: A schematic of the generators (both Landscape and Psychedelic).

Each of the generators (fig. 1) is a Convolutional Neural Network (CNN) having three components:

(i) **Encoder:** Takes as input a coloured image of resolution 256px by 256px and encodes it into a [64,64,256] feature vector by passing it through three convolutional layers.

(ii) **Transformer:** Uses six *resnet* blocks to transform the feature vector.

(iii) **Decoder:** Converts back the feature vector to a 256px by 256px coloured image by utilising two layers of transpose convolution.

Each of the discriminators (fig. 2) is a Convolutional Neural Network (CNN) with four layers, which takes as input an image and produces a one dimensional output denoting the probability associated with classifying whether the image is a generated one or not.

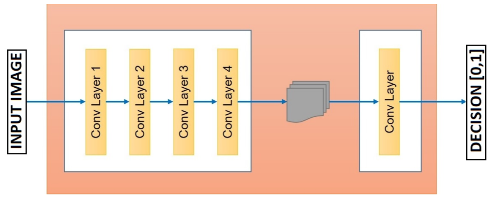

Figure 2: A schematic of the discriminators (both Landscape and Psychedelic).

2.3 Training the CycleGAN

Training of the CycleGAN consists of sequentially updating the weights of the generator and discriminator neural networks. To do so, various losses have been defined (fig. 3).

Loss functions for generators:

(i) *Psychedelic GAN loss* - Computed by taking a real landscape image and putting it through the PG to generate fake psychedelic art. The psychedelic art is then put through the PD. The output of the discriminator is then evaluated with binary cross-entropy loss which gives the Psychedelic GAN loss.

(ii) *Landscape GAN loss* - Computed similar to Psychedelic GAN loss, by taking real psychedelic art, putting it through the LG, then through the PD, and evaluating PD's output with binary cross-entropy loss.

(iii) *Landscape cycle-consistency loss* - Computed by putting fake psychedelic art through LG to generate the reconstructed landscape image. This reconstructed landscape image is evaluated against the real forest image with L1 loss which gives the Landscape cycle-consistency loss.

(iv) *Psychedelic cycle-consistency loss* - Computed similar to Landscape cycle-consistency loss, by putting fake landscape image through PG to generate the reconstructed psychedelic art, which is next evaluated against real psychedelic art with L1 loss.

(v) *Landscape identity loss* - Computed by putting the real landscape image through the LG and calculating the L1 loss by evaluating the real landscape image against the LG's output image which gives the Landscape identity loss.

(vi) *Psychedelic identity loss* - Computed similar to Landscape identity loss, by putting real psychedelic art through the PG and calculating the L1 loss by evaluating real psychedelic art against the PG's output image.

The *Generator loss* is the cumulative of all the above losses.
Generator loss = Psychedelic GAN loss + Landscape GAN loss + Landscape cycle-consistency loss + Psychedelic cycle-consistency loss + Landscape identity loss + Psychedelic identity loss

Loss functions for discriminators:

(i) *Landscape discriminator loss* - Computed by taking a real landscape image, putting it through LD, and evaluating the discriminator output with binary cross-entropy loss. Similarly, the previously generated fake landscape image is also put through the LD, and the discriminator output is evaluated with binary cross-entropy loss. The average of these two losses gives Landscape discriminator loss.

(ii) *Psychedelic discriminator loss* - Computed similar to Landscape discriminator loss by taking the average evaluated binary cross-entropy loss of the PD's output when real and fake psychedelic art are put through it.

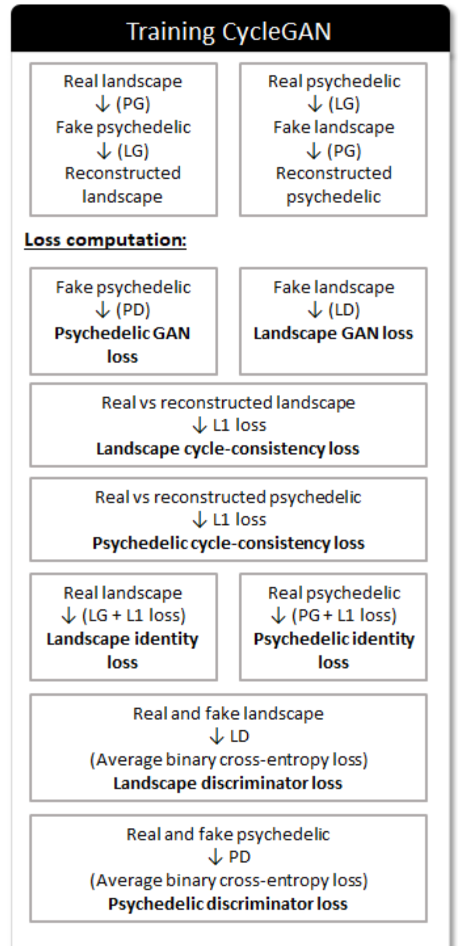

Figure 3: Schematic outlining different loss functions for generators and discriminators

Post loss calculation, stochastic gradient descent followed by backpropagation is employed to calculate the gradients and update the weights of the discriminator and generator neural networks with respect to these gradients. The Generator loss is used to update PG and LG. Landscape discriminator loss is used to update LD and Psychedelic discriminator loss is used to update PD. This process of loss calculation and updating the weights was carried out for 200 epochs (with a sample size of 10 images) whilst being trained on the Psychedelic and Landscape image data-sets (Section 2.1) each containing 150 images.

Figure 4: A sample of the original landscape images from the prepared Landscape image data-set

Figure 5: A sample of the original psychedelic art from the prepared Psychedelic art data-set

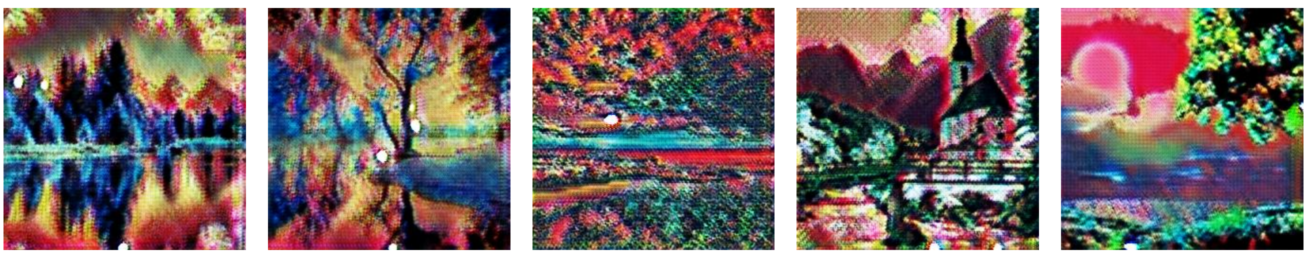

Figure 6: A sample of the artwork generated by PG during epochs 0 - 50

Figure 7: A sample of the artwork generated by PG during epochs 50 - 100

Figure 8: A sample of the artwork generated by PG during epochs 100 - 150

Figure 9: A sample of the artwork generated by PG during epochs 150 - 200

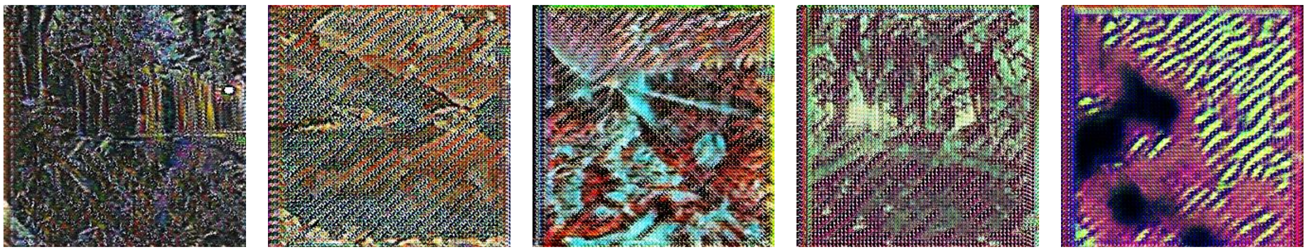

Figure 10: A sample of the texture containing artwork generated by PG post epoch count 150

Figure 11: A sample of the landscape images generated by LG

3 Results

As the epoch count varied during the training process, neural networks' learned weights also changed. Consequently, for different epoch counts, the PG generated different variants of psychedelic art. Though the loss was decreasing with the number of epochs, it was hard to objectively quantify and thus evaluate the visual appeal of the generated artwork corresponding to the epoch count. An unofficial poll among the students pursuing the course curriculum revealed the art-style learnt during epochs 100 - 150 (fig.4) as the most appealing. However, it is up to the reader to judge which style is most artistic or psychedelic.

3.1 Trend during training

Initially, as the training progressed and the number of epochs increased, the definition of the landscapes in the artworks generated by PG increased. Furthermore, in the beginning, the style of the generated artwork (while close to the art-style of the original psychedelic images) did not really go very well with the subject of landscapes. It is by around epoch count 150 that the artwork definition refined well enough. Post this epoch count, certain textures started appearing in the artwork (fig. 10) which reduced the visual appeal of the artwork.

3.2 Images generated by LG

While not the main objective, it is nevertheless interesting to review what LG generates (fig. 11). Contrary to PG which takes as input landscape images and generates psychedelic artwork (main objective), the LG takes as input psychedelic art and generates landscape images from it. This can be thought of as creating the subject of psychedelic art in the style acquired from landscapes. Since psychedelic art doesn't really have a concrete subject, rendering them in style of landscape images also yields images which may be best classified as abstract.

4 Conclusions

The goal of this study was to develop a framework for generating psychedelic art about landscapes. The art-style was to be learnt from psychedelic art available over the internet. To achieve this, a Cycle-Consistent Adversarial Network (CycleGAN) was developed and implemented to learn and carry out art generation from a data-set of unpaired content-style images. Content was landscape and the style was of psychedelic art. The network consisted of four neural networks - two generators (PG, LG) and two discriminators (PD, LD). The generators generate art/images on the basis of the style learnt and the discriminators judge whether an image is from the original data-set or it has been generated.

4.1 Key features of the architecture:

- All the individual neural networks were trained in tandem. For the training, custom loss functions were defined for generators and discriminators separately. Loss computation was followed by stochastic gradient descent and backpropagation steps to update the weights of the generator (PG, LG) and discriminator (PD, LD) neural networks.
- The Psychedelic Generator (PG) in the CycleGAN is responsible for learning the psychedelic art-style. IT takes as input 256px by 256px landscape images and outputs artwork created in the art-style learnt from psychedelic art found on the internet. For generating art post training, the landscape images can be sourced from anywhere and not just from the Landscape image data-set.
- The art-style learnt by the PG is determined by its weights which in-turn are determined by the amount of training (measured through epoch count) it has undergone. Thus, the learnt art-style can be referenced via the epoch count from during the training phase. Consequently, PG can create artwork in a specific art-style by setting its weights corresponding to the epoch count of style.

4.2 Scope of future work

The implemented framework was able to achieve the objective of generating psychedelic art. However, this architecture can further be refined and its functionality enhanced in some of the following ways:

- As the network is capable of generating artwork from landscape images sourced from anywhere, an application may be developed with the learnt weights of the neural network allowing a user to generate art from any landscape, perhaps even using an image from his/her own photo collection.
- Currently, the learnt art-style is catalogued via the epoch count corresponding to the weights of the network (learned during training phase). Art-styles with similar epoch counts are not necessarily similar. As a result, this cataloguing method is not very intuitive and user friendly. By grouping together similar art-styles, the ease of use of the network to generate art can be improved.
- The network has been developed to work with less powerful hardware. Consequently, as the image resolution of the landscape images and(or) psychedelic artwork increases, due to a multi-fold increase in computational cost, the performance of the network worsens owing to hardware limitations. However, if powerful hardware is available, the CycleGAN framework can be adjusted to exploit that powerful hardware. This would ultimately lead to generation of higher resolution artwork.

References

[Gatys *et al.*, 2015] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[Ng and Jordan, 2002] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.