## **Unpaired Image-to-Image Translation With CycleGAN**

A successful approach for unpaired image-to-image translation is CycleGAN.

CycleGAN is an approach to training image-to-image translation models using the generative adversarial network, or GAN, model architecture.

Traditionally, training an image-to-image translation model requires a dataset comprised of paired examples. That is, a large dataset of many examples of input images X (e.g. summer landscapes) and the same image with the desired modification that can be used as an expected output image Y (e.g. winter landscapes).

The requirement for a paired training dataset is a limitation. These datasets are challenging and expensive to prepare, e.g. photos of different scenes under different conditions.

The Cycle Generative adversarial Network, or CycleGAN for short, is a generator model for converting images from one domain to another domain.

For example, the model can be used to translate images of horses to images of zebras, or photographs of city landscapes at night to city landscapes during the day.

Image-to-image translation involves generating a new synthetic version of a given image with a specific modification, such as translating a summer landscape to winter.

The CycleGAN is a technique that involves the automatic training of image-to-image translation models without paired examples. The models are trained in an unsupervised manner using a collection of images from the source and target domain that do not need to be related in any way.

**Note**: CycleGAN technique for unpaired image-to-image translation.

**Basics:** The GAN architecture is an approach to training a model for image synthesis that is comprised of two models: a generator model and a discriminator model. The generator takes a point from a latent space as input and generates new plausible images from the domain, and the discriminator takes an image as input and predicts whether it is real (from a dataset) or fake (generated). Both models are trained in a game, such that the generator is updated to better fool the discriminator and the discriminator is updated to better detect generated images.

**Note:**

CycleGAN is an updated version of Pix2Pix GAN. CycleGAN uses unpaired image translation instead of paired image translation. This technique basically gives you a lot of opportunity to take any two pair images, and then transfer the properties of both the images to each other.

CycleGAN typically uses two generators and discriminators. The idea behind it is that image A is fed into the generator, and it generates a certain image G(A). The same image G(A) is fed into another generator to reconstruct the original image F(A). The name cycleGAN is inspired by the fact that instead of calculating the loss in a conventional way, cycleGAN calculates the loss of the ****original image**** and the ****reconstructed image****.

**CycleGAN architecture:**

The CycleGAN is an extension of the GAN architecture that involves the simultaneous training of two generator models and two discriminator models.

One generator takes images from the first domain as input and outputs images for the second domain, and the other generator takes images from the second domain as input and generates images for the first domain. Discriminator models are then used to determine how plausible the generated images are and update the generator models accordingly.

In other word, The CycleGAN uses an additional extension to the architecture called cycle consistency. This is the idea that an image output by the first generator could be used as input to the second generator and the output of the second generator should match the original image. The reverse is also true: that an output from the second generator can be fed as input to the first generator and the result should match the input to the second generator.

Eg:

Consider the problem where we are interested in translating images from summer to winter and winter to summer.

We have two collections of photographs and they are unpaired, meaning they are photos of different locations at different times; we don’t have the exact same scenes in winter and summer.

* ****Collection 1****: Photos of summer landscapes.
* ****Collection 2****: Photos of winter landscapes.

We will develop an architecture of two GANs, and each GAN has a discriminator and a generator model, meaning there are four models in total in the architecture.

The first GAN will generate photos of winter given photos of summer, and the second GAN will generate photos of summer given photos of winter.

* ****GAN 1****: Translates photos of summer (collection 1) to winter (collection 2).
* ****GAN 2****: Translates photos of winter (collection 2) to summer (collection 1).

Each GAN has a conditional generator model that will synthesize an image given an input image. And each GAN has a discriminator model to predict how likely the generated image is to have come from the target image collection. The discriminator and generator models for a GAN are trained under normal adversarial loss like a standard GAN model.

We can summarize the generator and discriminator models from GAN 1 as follows:

* ****Generator Model 1:****
  + ****Input****: Takes photos of summer (collection 1).
  + ****Output****: Generates photos of winter (collection 2).
* ****Discriminator Model 1****:
  + ****Input****: Takes photos of winter from collection 2 and output from Generator Model 1.
  + ****Output****: Likelihood of image is from collection 2.

Similarly, we can summarize the generator and discriminator models from GAN 2 as follows:

* ****Generator Model 2****:
  + ****Input****: Takes photos of winter (collection 2).
  + ****Output****: Generates photos of summer (collection 1).
* ****Discriminator Model 2****:
  + ****Input****: Takes photos of summer from collection 1 and output from Generator Model 2.
  + ****Output****: Likelihood of image is from collection 1.

**Architecture explanantion:**

The architecture is comprised of four models, two discriminator models, and two generator models.

**In cycleGAN,**

**Descriminator model has DCGAN architecture.**

**Generator is an encoder-decoder model architecture**. The model takes a source image (e.g. horse photo) and generates a target image (e.g. zebra photo). It does this by first downsampling or encoding the input image down to a bottleneck layer, then interpreting the encoding with a number of ResNet layers that use skip connections, followed by a series of layers that upsample or decode the representation to the size of the output image.

### **Building the generator**

### *IMG_256*

The generator have three components:

1. Encoder
2. Transformer
3. Decoder

**Applications:**

* **Collection Style Transfer:** The authors trained the model on landscape photographs downloaded from Flickr and WikiArt. Unlike other works on neural style transfer, CycleGAN learns to mimic the style of an entire collection of artworks, rather than transferring the style of a single selected piece of art. Therefore it can generate different  styles such as : Van Gogh, Cezanne, Monet, and Ukiyo-e

**Object Transformation**: CycleGAN can transform object from one ImageNet class to another such as: Zebra to Horses and vice-versa, Apples to Oranges and vice versa etc.

* **Season Transfer**: CycleGAN can also transfer images from Winter Season to Summer season and vice-versa. For this the model is trained on 854 winter photos and 1273 summer photos of Yosemite from Flickr.
* **Photo Generation from Painting**: CycleGAN can also be used to transform photo from paintings and vice-versa. However to improve this transformation., the authors also introduced an additional loss called Identity loss. This loss can be defined as :
* **Photo enhancement** : CycleGAN can also be used for photo enhancement. For this the model takes images from two categories which are captured from smartphone camera (usually have deep Depth of Field due to low aperture ) to DSLR (which have lower depth of Field). For this task the model transforms images from smartphone to DSLR quality images.

Refernce:

<https://machinelearningmastery.com/what-is-cyclegan/>