

Architecture & Dataset Report

Srikrishna Dantu

19MCME02

Shobhit Kumar

19MCME16

KPSSS Srinu

19MCME26

Project Title: Converting Photos Into Paintings Of A Specific Style

Our project belongs to the domain where we need to generate an image (painting) given another image (captured photo) (image-to-image translation). Many architectures are available to do this work; we have listed a few of the architectures which can be useful for our project.

We chose CycleGAN as the architecture for our project. We described the reasons to chose this architecture among several other architectures in detail in the below context.

Reference to the original CycleGAN[\[2\]](#) paper for details.

Architecture that we can use:

Encoder-Decoder Network

We need to have pair of input and output pictures where the input will be a photo taken from the camera, and the output will be painting in that specific style what is in the photo.

Getting a bunch of photos is easy, but we cannot have paintings of those exact styles in photos. Vice-versa is also not possible; we can have a bunch of paintings, but getting photos of the entity in a similar environment is not easy.

So the encoder-decoder network is not best suited for our project.

Neural Style Transfer

Neural style transfer is another famous way to perform image-to-image translation, which synthesizes an image by combining the content of one image with the style of another image based on matching the *Gram matrix statistics* of pre-trained deep features (VGG-19). But it performs the

translation between two specific images. What we need is to focus on the mapping between two image collections and capture correspondences between the higher-level appearance structure (color, texture, etc.). Refer to the original paper[1] for details.

CycleGAN

Since, for us, getting paired training data is not possible. Cycle-consistent adversarial networks (CycleGAN)[2] translate an image from a source domain X to a target domain Y in the absence of paired training examples. The architecture assumes that there is some underlying relationship between both domains.

Although we lack supervision in the form of paired examples, the architecture exploits supervision at the level of sets: given one set of images in the domain X and a different set of images in the domain Y . We train a mapping $G : X \rightarrow Y$ such that the output $\hat{y} = G(x)$, $x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify \hat{y} apart from y .

Why is CycleGAN cyclic in architecture?

In practice, it is found that it is difficult to optimize the adversarial objective in isolation: standard procedures often lead to the well-known problem of *mode collapse*, where all input images map to the same output image, and the optimization fails to make progress.

Therefore, the architecture exploits the property that translation should be “*cycle consistent*”. Mathematically, if we have a translator $G : X \rightarrow Y$ and another translator $F : Y \rightarrow X$, then G and F , should be inverse of each other, and both mapping should be bijections.

The architecture applies this structural assumption by training both mapping G and F simultaneously and adding a “*cycle consistency loss*” that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Combining this loss with

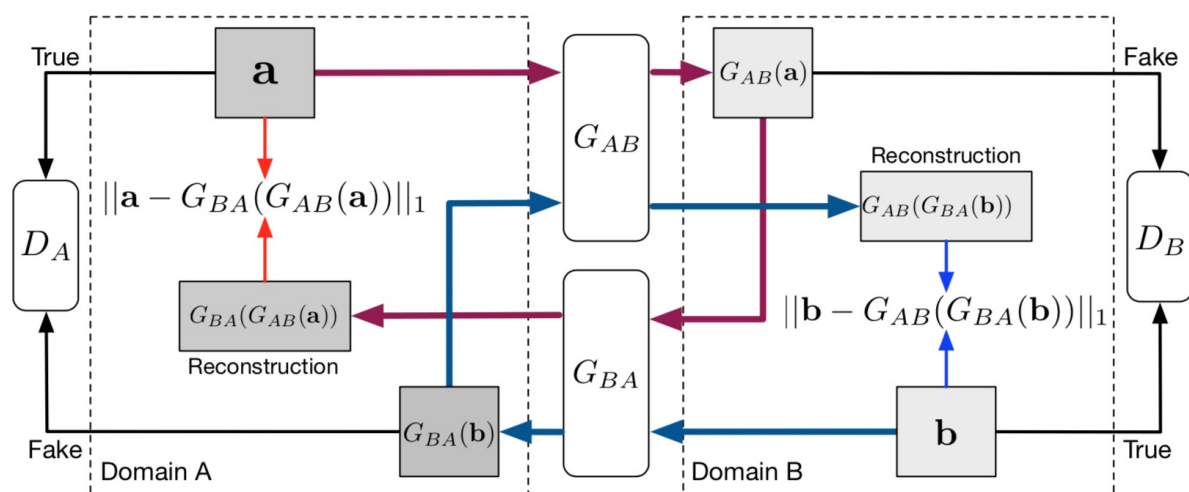
“adversarial losses” on domain X and Y yields the full objective for unpaired image-to-image translation.

[“Adversial Loss” & “Cycle Consistency Loss” to be discussed in Loss Function Report]

Refer [3] & [4] for easy explanation on CycleGAN.

Network Architecture

The architecture has two generators (G and F) and their corresponding discriminator.



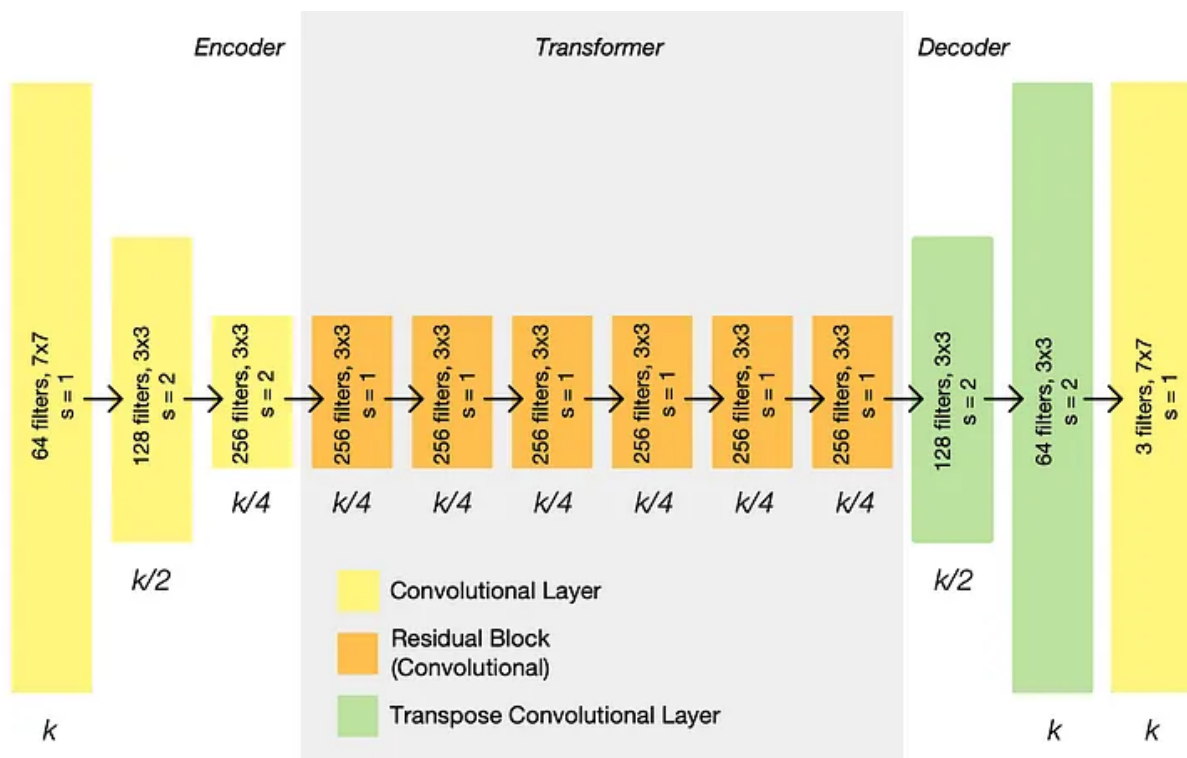
CycleGAN Network Architecture | Image Credits: imgur.com

As seen in the above figure,

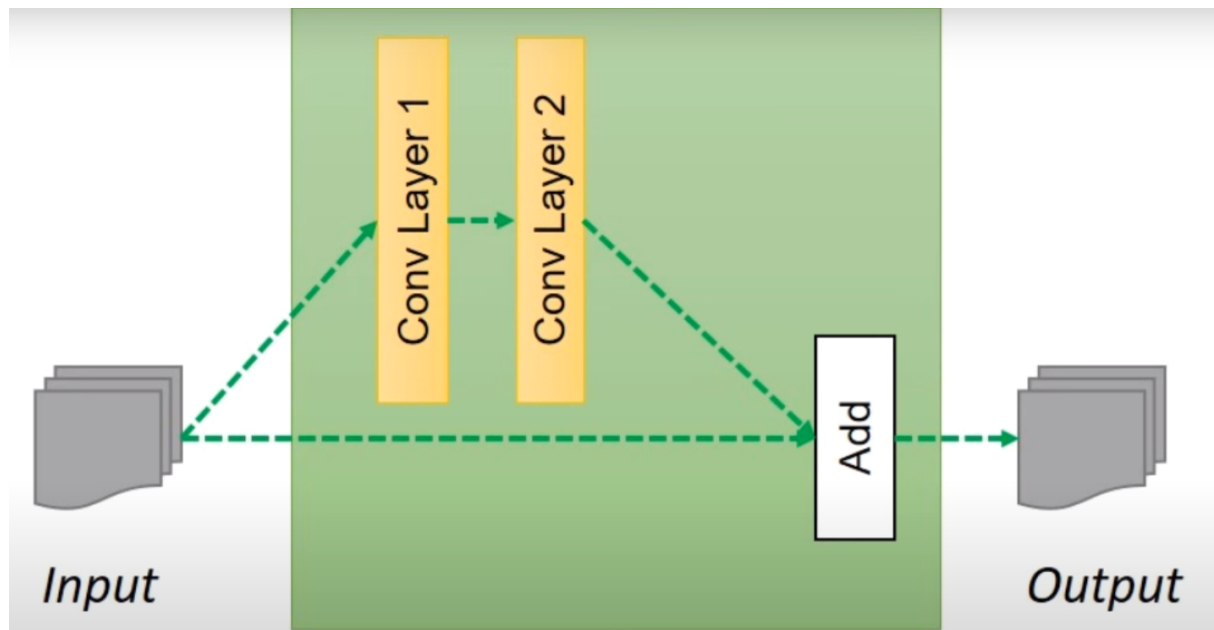
- We have two different domains of images (A & B), with the image (a & b) as true images.
- G_{AB} and G_{BA} are two *generators* that translate the image from the domain A to B and the image from the domain B to A simultaneously.
- D_A and D_B are two *discriminators* that will discriminate true a with fake generated \hat{a} or ($G_{BA}(b)$) and true b with fake generated \hat{b} or ($G_{AB}(a)$) simultaneously and will output binary (0 or 1) for fake or real.

Generator

- 6 residual blocks for 128x128 training image
- 9 residual blocks for 256x256 or higher resolution training image
- C7s1-k \rightarrow 7x7 Convolution-InstanceNorm-ReLU layer with k-filters and stride 1
- dk \rightarrow 3x3 Convolution-InstanceNorm-ReLU layer with k-filters and stride 2
- uk \rightarrow 3x3 Convolution-InstanceNorm-ReLU layer with k-filters and stride $\frac{1}{2}$
- Rk \rightarrow Residual block contains 3x3 Convolution layer with same no. of filters on both layers
- Network with 6 residual blocks
 - C7s1-32, d64, d128, 6*(R128), u64, u32, C7s1-3
- Network with 9 residual blocks
 - C7s1-32, d64, d128, 9*(R128), u64, u32, C7s1-3



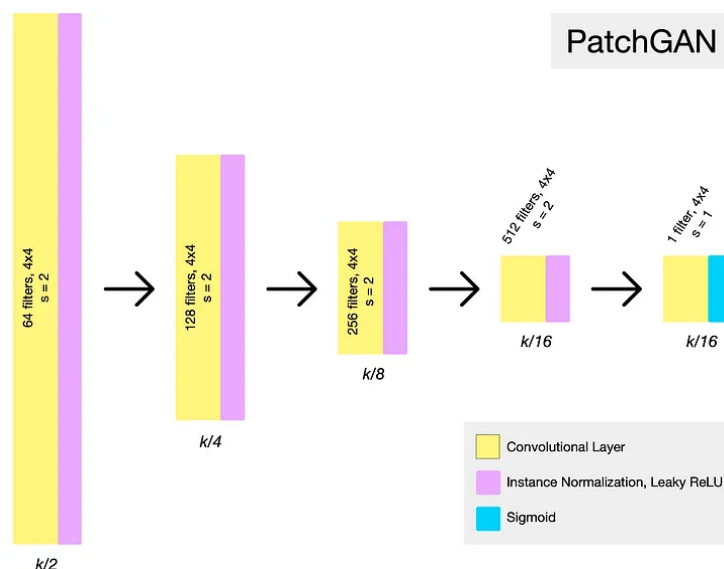
Generator Architecture for 6 ResNet Block | Image Credits: [Sarah Wolf - Toward Data Science](#)



Residual Block Architecture | Image Credits: [CodeEmporium](#)

Discriminator

- 70x70 PatchGAN
- $C_k \rightarrow$ 4x4 Convolution-InstanceNorm-ReLU layer with k -filters and stride 2
- Discriminator Network
 - C64, C128, C256, C512
- After the last layer, apply a Convolution to produce 1-dimensional output
- Do not use InstanceNorm for the first C64 layer
- Use Leaky-ReLU with slope 0.2




Discriminator Architecture | Image Credits: [Sarah Wolf - Toward Data Science](#)

DATASETS

We are following two datasets (X & Y domains), further in Y domain, we are considering two different artists Vincent Van Gugh[6] and Monet[5].

We will first work with the Vincent Van Gugh dataset, and later if resources & time permit, we will also work with the Monet dataset. And compare both results.

References

- [1] [\[1508.06576\] A Neural Algorithm of Artistic Style](#)
- [2] [\[1703.10593\] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks](#)
- [3] [CycleGAN: Learning to Translate Images \(Without Paired Training Data\) | by Sarah Wolf](#)
- [4]  [Unpaired Image-Image Translation using CycleGANs](#)
- [5] [I'm Something of a Painter Myself | Kaggle](#)
- [6] [Best Artworks of All Time | Kaggle](#)