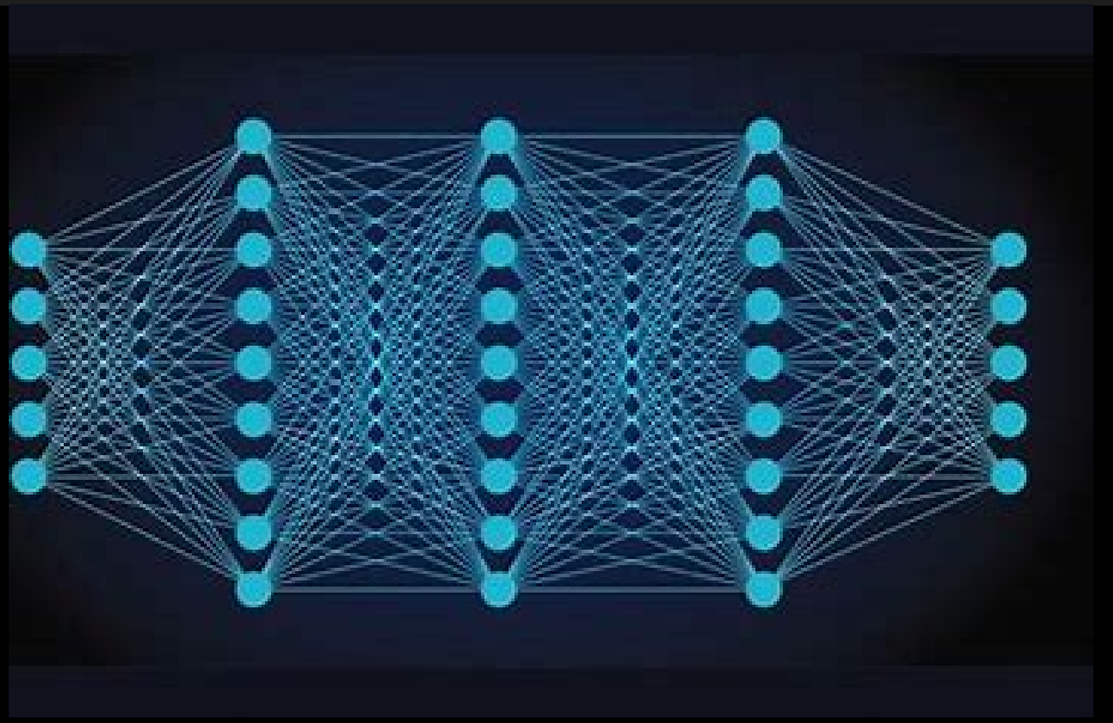
Artificial Intelligence 101

Research Paper-02 Presentation on

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks



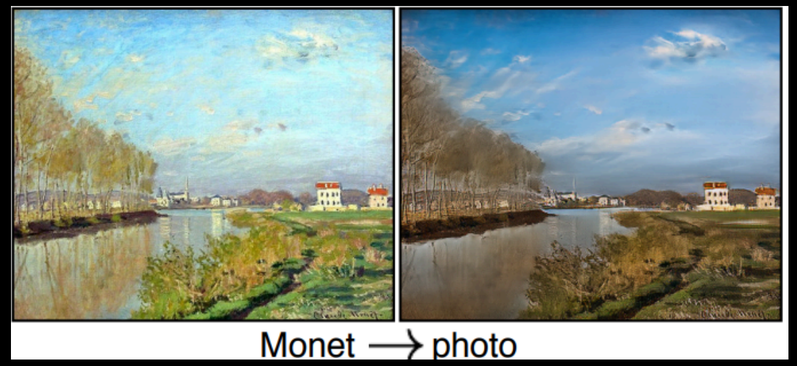
Watt Wizards
Group - 6

Introduction

In this paper, we introduce a method for unpaired image-to-image translation, where we learn to translate images from one domain (e.g., Monet paintings) to another (e.g., photographs) without requiring paired training examples.

Motivation:
Imagine converting a Monet painting to a photograph of the same scene, capturing the unique stylistic differences without paired examples. This is a challenging problem but has wide applications in fields like artistic stylization, semantic segmentation, and object transfiguration.

Challenge:
Supervised models often require paired examples (e.g., a photograph and its corresponding painting). However, obtaining such pairs is difficult and expensive.



What already existed?

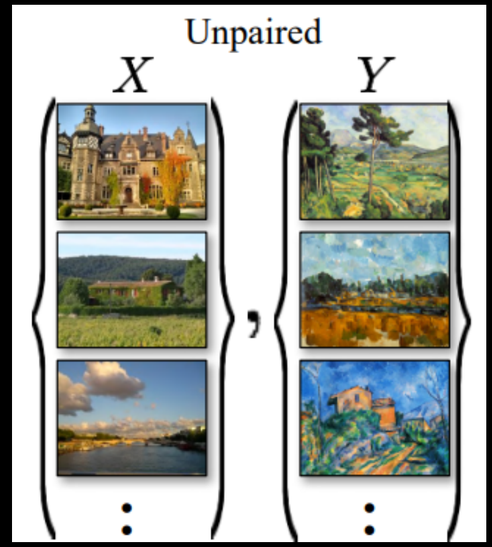
The model works on Image-to-image translation which is a class of vision and graphics problems.

Similar models were built where the goal was to learn the mapping between an input image and an output image using a training set of aligned image pairs.

What was the problem?

The model focussed on an approach to translate an image from a source domain to a target domain in the absence of paired examples. This was because, for many tasks, paired training data was not be available.

It also introduced adversial loss and cycle consistency loss together which was something new.



WHAT IS GAN?

GAN stands for Generative Adversarial Network, a deep learning neural network consists of two models Generator and Discriminator that compete with each other to analyze, capture and copy the variations within a dataset

Types of GANs:

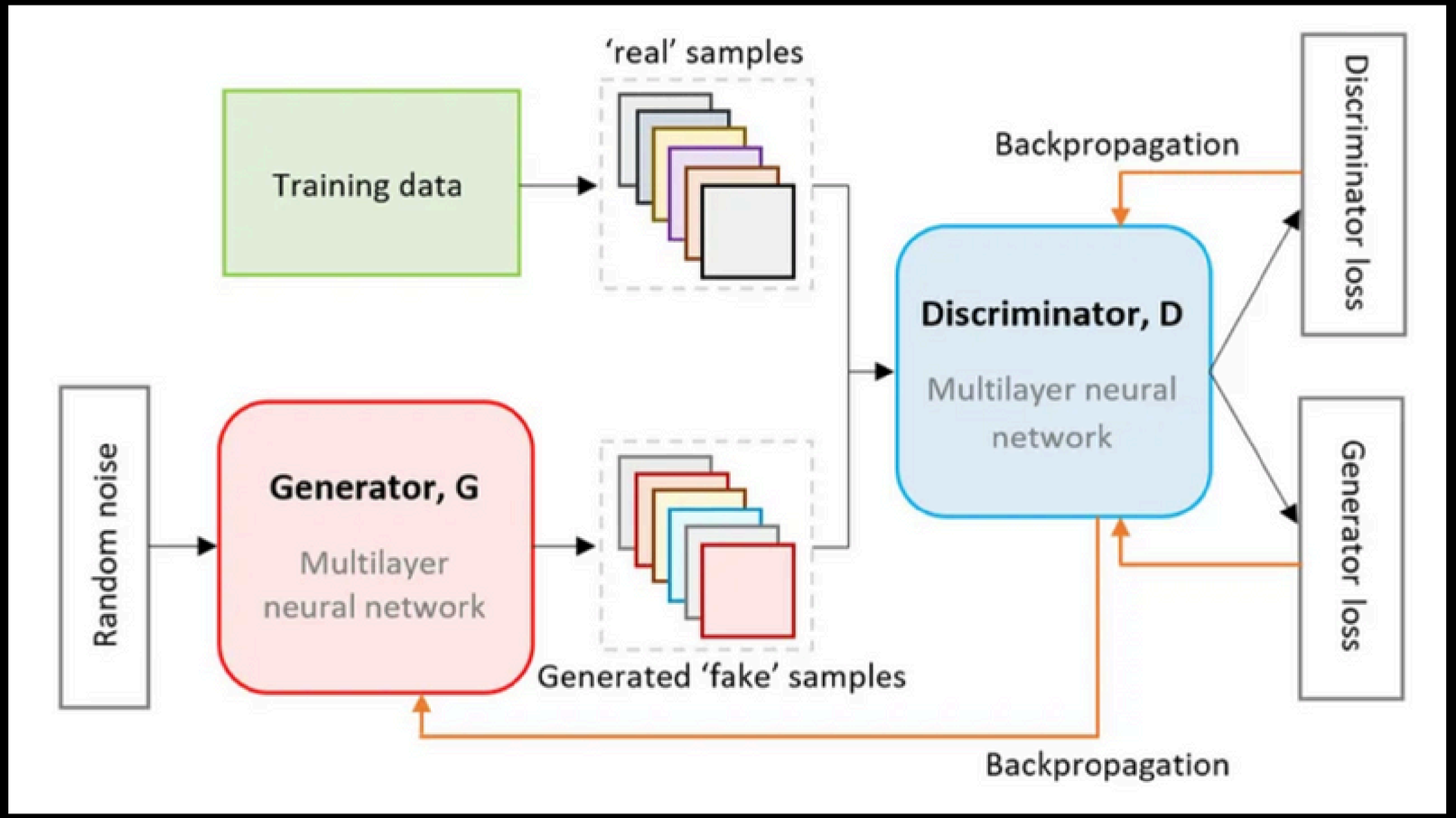
1.Vanilla GAN

2.Conditional GAN (CGAN)

3.Deep Convolutional GAN (DCGAN)

4.Laplacian Pyramid GAN (LAPGAN)

5.Super Resolution GAN (SRGAN)



Related Approaches and why they have failed

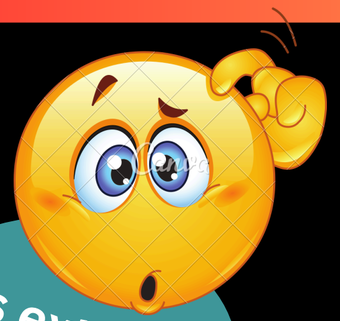
1.Paired Image-to-Image Translation (e.g., pix2pix)

Approach

Uses paired data to train models for tasks like labels-to-photos or sketch-to-image

GAN Use: :

Uses a **Conditional GAN** to translate images when we have matching pairs of input and output images.



Requires extensive paired datasets, which are expensive or impractical to gather for many tasks.

Why Failed Then??

What Could be explored!!!

Methods to leverage unpaired data more effectively, as paired datasets aren't scalable for all domains.

2.CoGAN and other associated GANs:-

GAN Use:

Utilizes Coupled GANs where two GANs share weights, enabling the learning of joint representations across different domains effectively.

Each GAN is responsible for handling one specific domain, allowing the system to capture and transfer knowledge between the two domains seamlessly.

Why It Fails:

It uses weight sharing to enforce consistency, but this approach doesn't effectively preserve the content structure during translation, limiting its performance in maintaining the original image's content across domains.

Learning Mappings Between Two Domains

Objective: Learn mapping functions between two domains X and Y using unpaired training data:

- **Training Data:**
  - $\{x_i\}_{i=1}^{N}$, where $x_i \in X$ (source domain)
  - $\{y_j\}_{j=1}^{M}$, where $y_j \in Y$ (target domain)

- **Data Distributions:**
  - $x \sim p_{\text{data}}(x)$
  - $y \sim p_{\text{data}}(y)$

2. Model Structure Mappings:

G:X→Y (transforms from source domain to target domain)

F:Y→X (transforms from target domain to source domain)

Adversarial Discriminators:

DX: Discriminates between real images from domain X and fake images F(y)

DY: Discriminates between real images from domain Y and fake images G(x)

3. Objective Function

Loss Types:

Adversarial Losses: Match the distribution of generated images to the real data distribution in the target domain.

Cycle Consistency Losses: Ensure that the mappings G and F do not contradict each other.

Cycle consistency: If x is mapped to y and then back to x, the result should resemble the original x, and similarly for y.

Adversarial Loss

Goal: Generate images that resemble the target domain, Y, while distinguishing them from real target domain samples.

For the mapping G:X→Y: $L_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]$

- Generator G aims to create images that look like real images from Y.
- Discriminator DY tries to distinguish between real and fake images (from G(x)).

Objective: Minimize L(GAN) for G and maximize for DY, i.e. $\min_G \max_{D_Y} L_{\text{GAN}}(G, D_Y, X, Y).$

Same process for mapping F:Y->X

Cycle Consistency Loss

Goal: Ensure that when an image is translated to the target domain and then back, it should resemble the original image.

Forward Cycle Consistency:
x→G(x)→F(G(x))≈x The image x from domain X should be reconstructed after two mappings.

Backward Cycle Consistency:
y→F(y)→G(F(y))≈y
Similarly, the image y from domain Y should return to the original after two mappings.

Cycle Consistency Observations:
The reconstructed images F(G(x)) closely match the input images x, as shown in experiments like photo ⇔ Cezanne, horses ⇔ zebras, and more.

Cycle Consistency Loss: $L_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$

Ensures that the mappings G and F do not contradict each other and preserve input-output relationships.

Full Objective

Objective Function

$$L(G, F, D_X, D_Y) = L_{\text{GAN}}(G, D_Y, X, Y) + L_{\text{GAN}}(F, D_X, Y, X) + \lambda L_{\text{cyc}}(G, F)$$

Adversarial Loss: (LGAN) for both mappings.
Cycle Consistency Loss: L(cyc) ensures forward and backward consistency.
λ: A hyperparameter, balances the losses.

Optimization

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} L(G, F, D_X, D_Y)$$

Generators are minimized, discriminators are maximized.

Empirical Results:

Both adversarial and cycle consistency losses are crucial for high-quality results.
Single cycle is insufficient for regularization.

Autoencoder Analogy

Two autoencoders: F∘G:X→X and G∘F:Y→Y
Adversarial autoencoders regularize the intermediate representation via adversarial loss.

Network Architecture

Generative Network Architecture

- Architecture for generative networks is adopted from Johnson et al.
- The network consists of three convolutional layers, followed by several residual blocks.
- It includes two fractionally-strided convolutions with a stride of 0.5.
- For images of size 128x128, 6 residual blocks are used.
- For 256x256 or higher-resolution images, 9 blocks are used.
- A final convolution layer maps the feature maps to RGB output.

Discriminator Network Architecture

- The discriminator networks use 70x70 PatchGANs.
- PatchGANs are designed to classify whether 70x70 overlapping image patches are real or fake.
- This patch-based discriminator architecture has fewer parameters compared to a full-image discriminator.
- It operates in a fully convolutional manner, allowing it to work with images of any size.

Training Details

Two techniques are applied to stabalize model training procedure.

Firstly, the model use least-squares loss, this loss is more stable (than negative log-likelihood) during training and generate higher quality results. We train G to minimize LG and D to minimize LD.

$$L_G = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[(D(G(x)) - 1)^2 \right]$$

$$L_D = \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[(D(y) - 1)^2 \right] + \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[(D(G(x)))^2 \right]$$

Secondly, to reduce model oscillation the model uses Shrivastava et al.'s strategy. Discriminators are updated using a history of generated images rather than the ones produced by the latest generators. The model keeps an image buffer that stores the previously created images.

CycleGAN Applications and Results

1. Collection Style Transfer

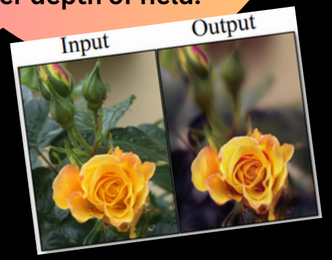
Goal: Generate photos in the style of an entire collection rather than a single artwork.
Dataset: Landscape photos from Flickr and WikiArt
Key Advantage: Learns broader style features from multiple artworks, not just one.

2. Object Transfiguration

Goal: Translate between visually similar object categories (e.g., horse to zebra).
Dataset: ImageNet object classes with ~1000 images per category.
Key Difference: Focuses on transfiguration between categories that are visually similar.

5. Photo Enhancement (Shallower Depth of Field)

Goal: Generate photos with shallower depth of field from photos with deep depth of field.
Dataset: Flower photos from Flickr.
Source Domain: Smartphone photos with deep depth of field (small aperture).
Target Domain: DSLR photos with shallow depth of field (larger aperture).
Key Feature: The model successfully transforms smartphone photos into DSLR-like images with shallower depth of field.



4. Photo Generation from Paintings

Goal: Generate realistic photos from paintings.
Key Technique: Introduce a loss function to preserve color composition between input (painting) and output (photo).
Identity Loss: Encourages the generator to resemble real photos when fed with target domain data.

3. Season Transfer

Goal: Convert images of Yosemite National Park from winter to summer (or vice versa).
Dataset: 854 winter photos, 1273 summer photos (from Flickr).
Key Feature: Transforms landscapes based on seasonal differences (e.g., snow, vegetation).

Limitations and Discussion

Successes: Method works well for tasks involving color/texture changes

Challenges: Generator architecture is optimized for appearance, not geometry. Geometric transformations remain a key challenge. (e.g., dog → cat).

Dataset Limitations: Misclassifications occur due to dataset inconsistencies (e.g., horse → zebra).

Paired vs. Unpaired Data: Unpaired data lags behind paired training methods, leading to issues like label permutation (tree vs. building).

Thank You!