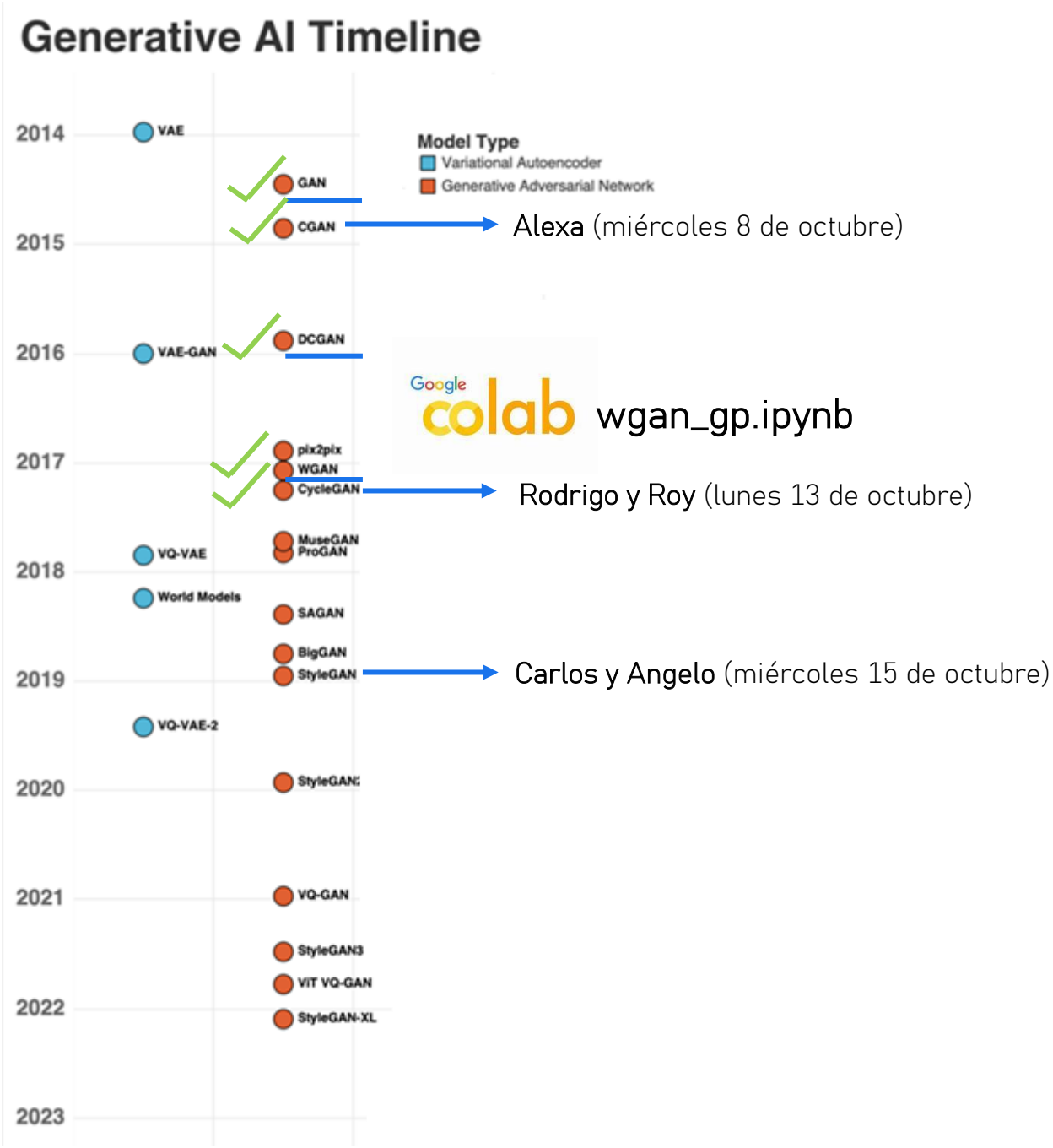
GANs IV

Clase 15

Dra. Wendy Aguilar

Modelos Generativos Profundos

UN ENFOQUE DESDE LA CREATIVIDAD COMPUTACIONAL

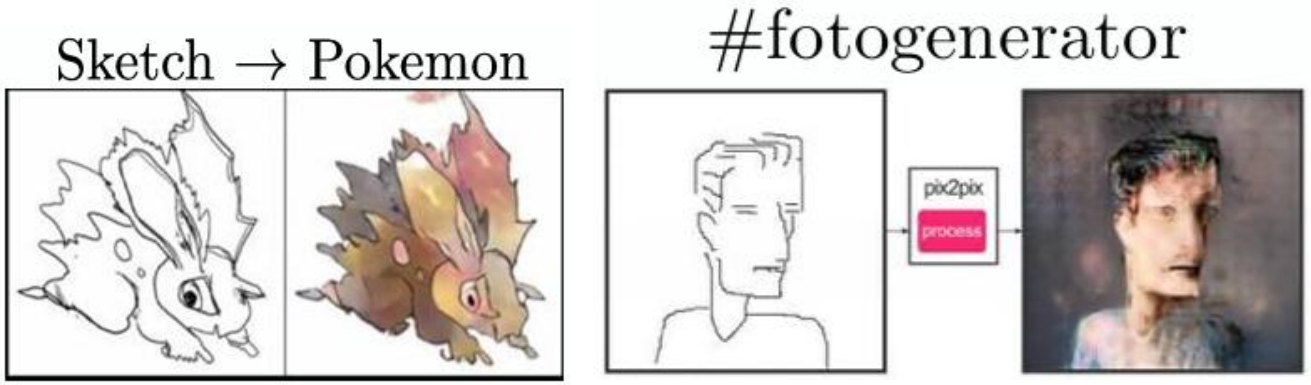

Generative AI Timeline
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
VAE
GAN
CGAN
VAE-GAN
DCGAN
pix2pix
WGAN
CycleGAN
VQ-VAE
MuseGAN
ProGAN
World Models
SAGAN
BigGAN
StyleGAN
VQ-VAE-2
StyleGAN2
VQ-GAN
StyleGAN3
ViT VQ-GAN
StyleGAN-XL
Model Type
Variational Autoencoder
Generative Adversarial Network
Alexa (miércoles 8 de octubre)
Google colab wgan_gp.ipynb
Rodrigo y Roy (lunes 13 de octubre)
Carlos y Angelo (miércoles 15 de octubre)

CycleGAN
¿Qué problema resolvieron?

CycleGAN

Tarea de **traducción de imágenes** (image-to-image translation)

- Idealmente se tiene un **dataset pareado**:
  cada imagen del dominio A tiene su **contraparte exacta** en el dominio B.



En este caso, cada par (x_i, y_i) representa **la misma escena o contenido**, pero en dos estilos diferentes.

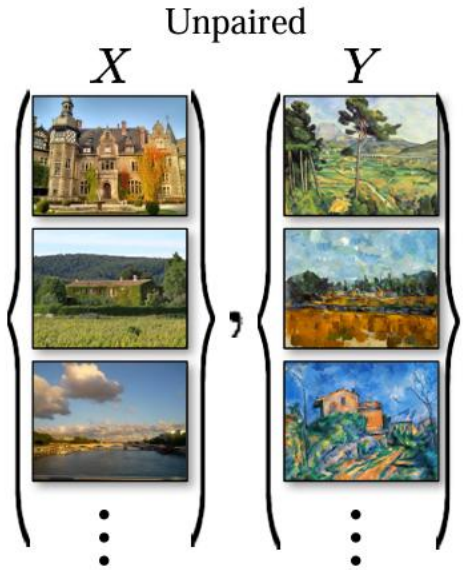
- Este tipo de datasets pareados son los que usan modelos como Pix2Pix.

CycleGAN

Tarea de **traducción de imágenes** (image-to-image translation)

- CycleGAN fue diseñada para situaciones donde **solo tenemos colecciones independientes** de imágenes de ambos dominios, pero **no hay pares que correspondan entre sí**:

Conjunto X (dominio fuente)	Conjunto Y (dominio destino)
Fotos de caballos	Fotos de cebras
Fotos en verano	Fotos en invierno
Retratos reales	Pinturas al óleo



Y queremos traducir por ejemplo:

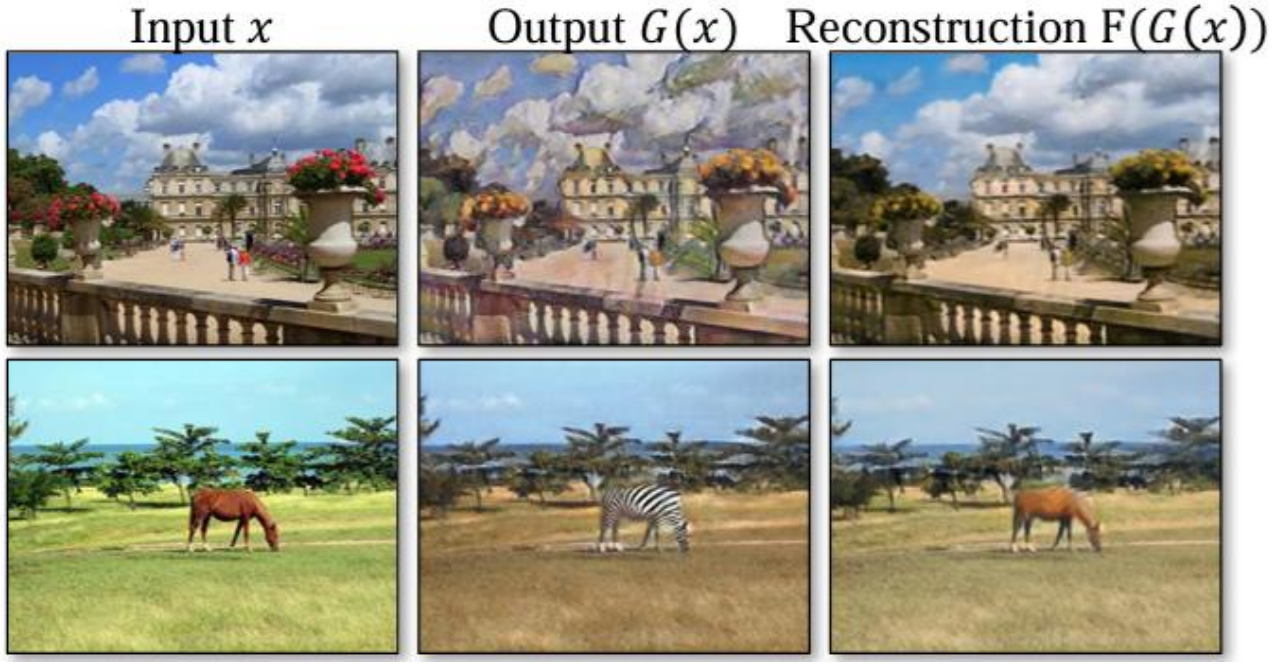
CycleGAN

¿Qué estrategia utilizaron para resolver la falta de pares?

CycleGAN

- Para resolver este problema de falta de pares, la CycleGAN introduce la **pérdida de consistencia de ciclo**:

$$F(G(x)) \approx x \quad \text{y} \quad G(F(y)) \approx y$$



- Si traduzco un caballo a cebra y luego regreso a caballo, debería obtener un caballo parecido al original.

- Esa *reversibilidad* actúa como una forma de **auto-supervisión**, compensando la ausencia de pares reales.

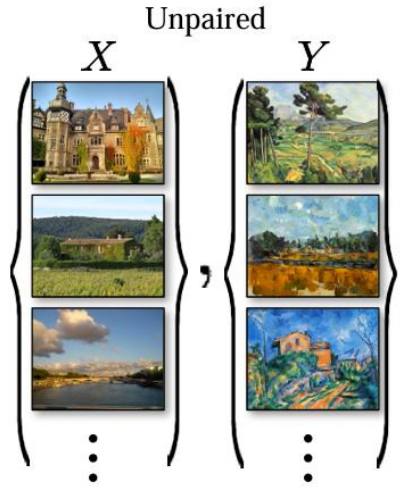
CycleGAN

¿Cuántos generadores y cuántos discriminadores tiene su arquitectura?

CycleGAN

Arquitectura general

CycleGAN está compuesto por **dos generadores** y **dos discriminadores**:



fotos de *paisajes*

pinturas al estilo van Gogh

Dos generadores aprenden mapeos opuestos:

- $G: X \rightarrow Y$ (traduce fotos de paisajes en pintura estilo Monet)
- $F: Y \rightarrow X$ (traduce pinturas estilo Monet a fotos de paisajes)

Cada generador tiene su discriminador correspondiente:

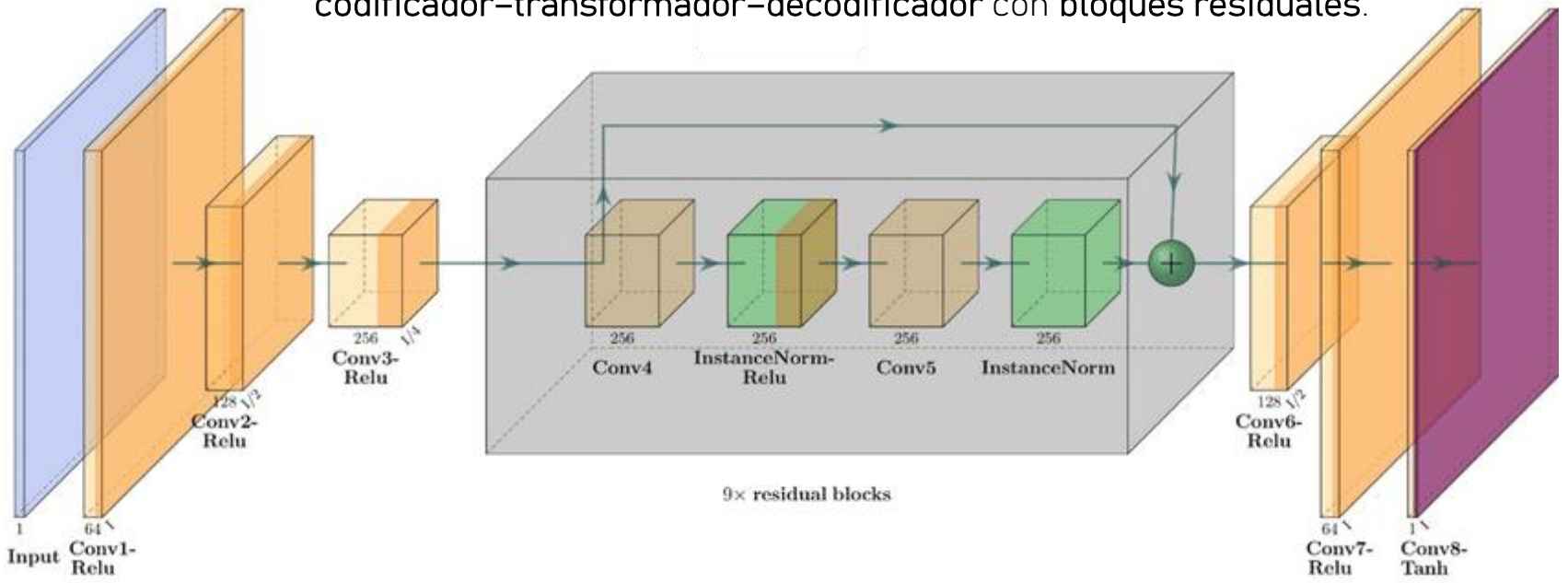
- D_Y : distingue imágenes reales de Y vs. las falsas de $G(X)$.
- D_X : distingue imágenes reales de X vs. las falsas de $F(Y)$.

CycleGAN

¿Cómo es la arquitectura de los generadores?

Arquitectura de los Generadores de la CycleGAN

Red convolucional profunda (DCNN),
compuesta por múltiples capas convolucionales organizadas en un esquema
codificador–transformador–decodificador con **bloques residuales**.



"Resume" la imagen y extrae lo más importante antes de transformarla

Cambia la "apariencia" pero conserva la "escena"

"Pinta la nueva imagen" a tamaño completo, con el nuevo estilo.

$$\text{Imagen} \xrightarrow{\text{Conv (encoder)}} \text{Features jerárquicas} \xrightarrow{\text{ResNet blocks}} \text{Transformación de estilo} \xrightarrow{\text{Deconv (decoder)}} \text{Imagen traducida}$$

Extrae la información importante de la imagen y **reduce su tamaño** para concentrarse en las estructuras esenciales (forma, composición, bordes).

Cada bloque residual aplica convoluciones que alteran texturas, colores, iluminación, pinceladas, etc.

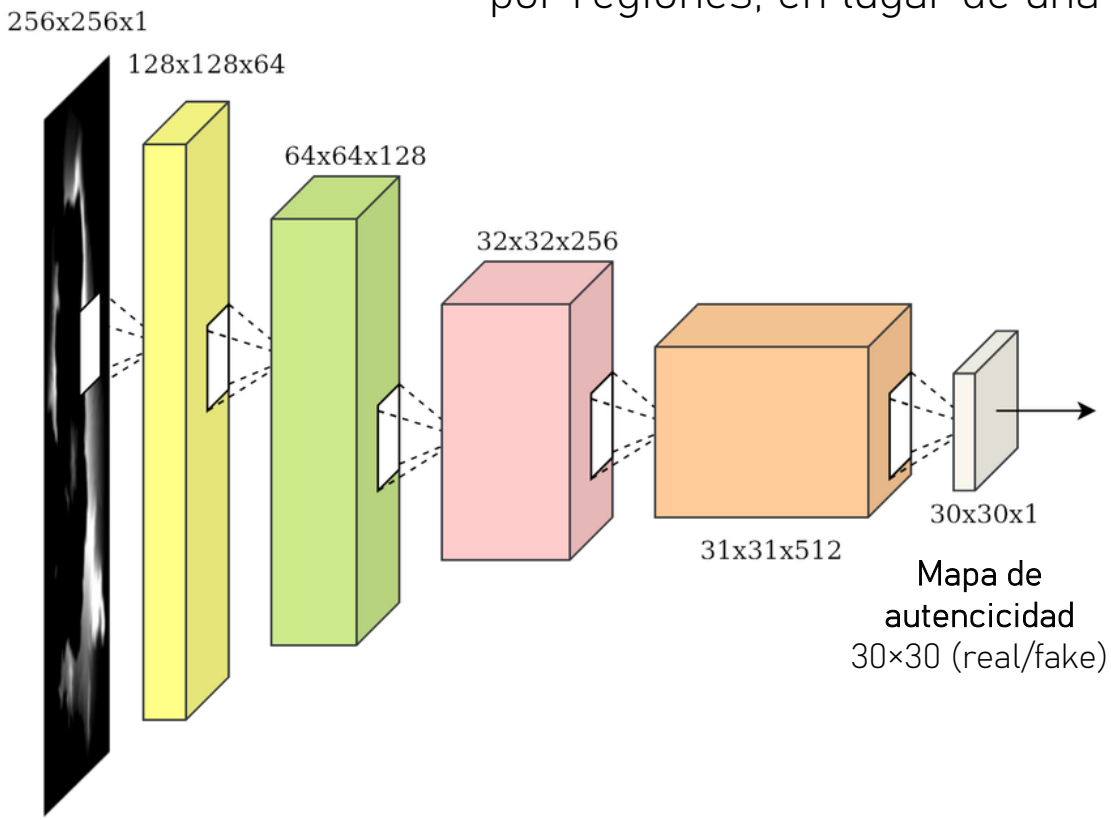
Toma las características transformadas (ya con el nuevo estilo) y las **vuelve a convertir en una imagen completa**, restaurando la resolución original.

CycleGAN

¿Cómo es la arquitectura de los discriminadores?

Arquitectura de los Discriminadores de la CycleGAN

Es un **discriminador PatchGAN**
con cuatro bloques convolucionales (C64–C128–C256–C512)
que analiza parches locales de la imagen y produce un mapa de autenticidad
por regiones, en lugar de una sola predicción global.



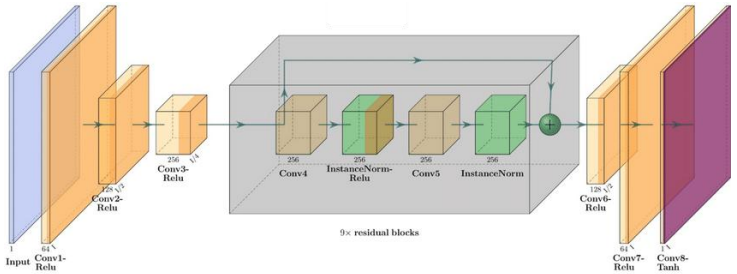
Mapa de autencicidad
30×30 (real/fake)

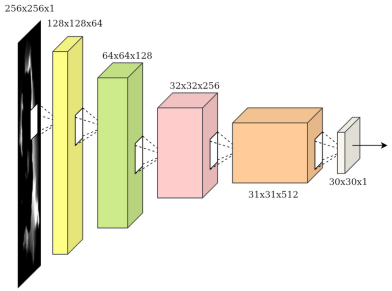
Cada celda de ese mapa corresponde a **un parche de la imagen original** —con un campo receptivo de aproximadamente **70×70 píxeles**— y su valor indica qué tan *real* parece **esa región local**.

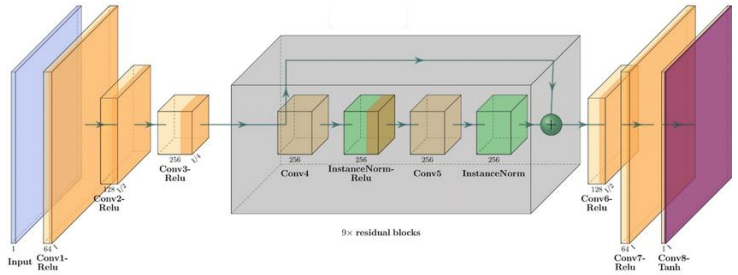
CycleGAN

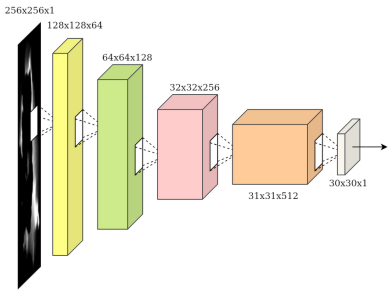
¿Cómo se integran los dos generadores y los dos discriminadores en una sola arquitectura coherente?

¿De qué manera compiten y cooperan simultáneamente?





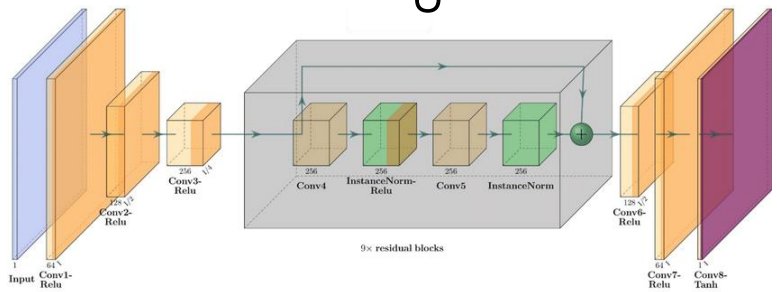




1. Los generadores están conectados en ciclo:

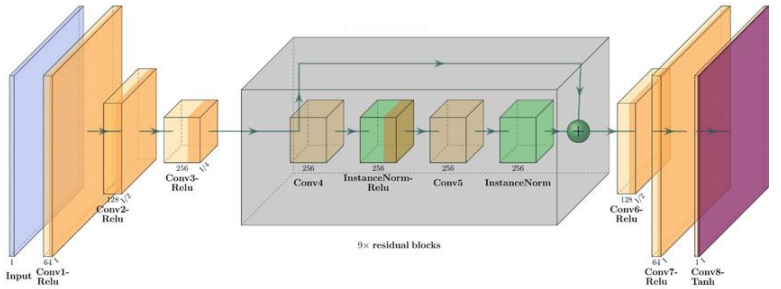
- $G: X \rightarrow Y$ (traduce fotos de paisajes en pintura estilo Monet)
- $F: Y \rightarrow X$ (traduce pinturas estilo Monet a fotos de paisajes)

G

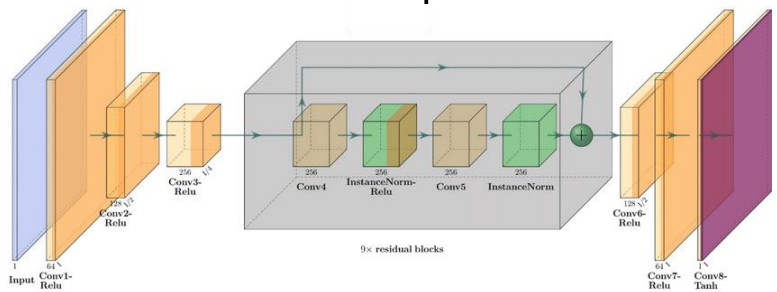


Ciclo 1

F

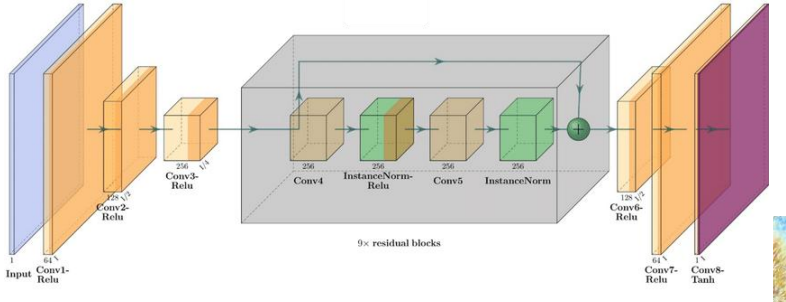


F



Ciclo2

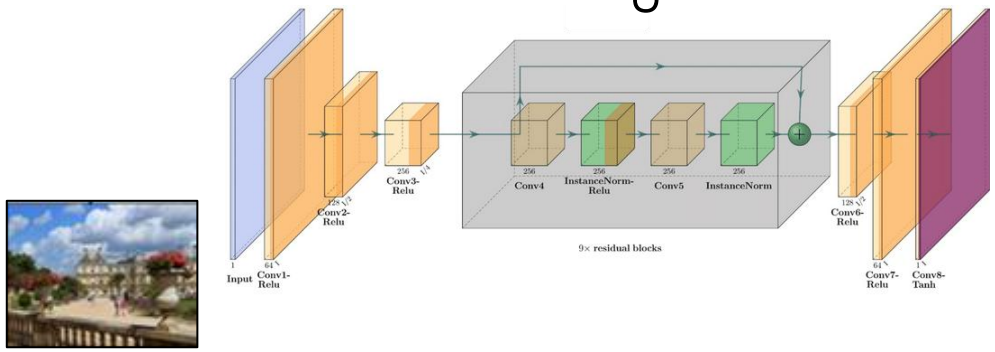
G



1. Los generadores están conectados en ciclo:

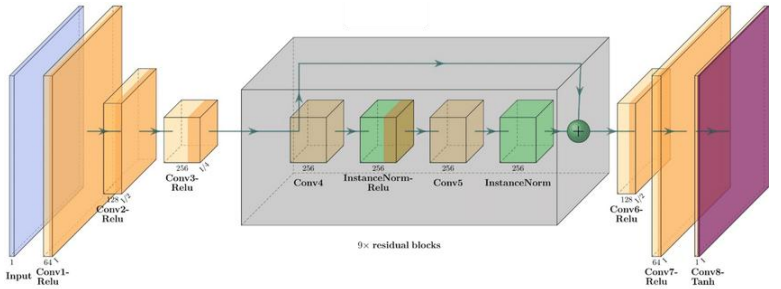
- $G: X \rightarrow Y$ (traduce fotos de paisajes en pintura estilo Monet)
- $F: Y \rightarrow X$ (traduce pinturas estilo Monet a fotos de paisajes)

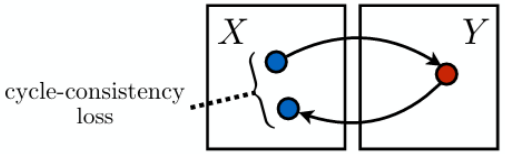
G



Ciclo 1

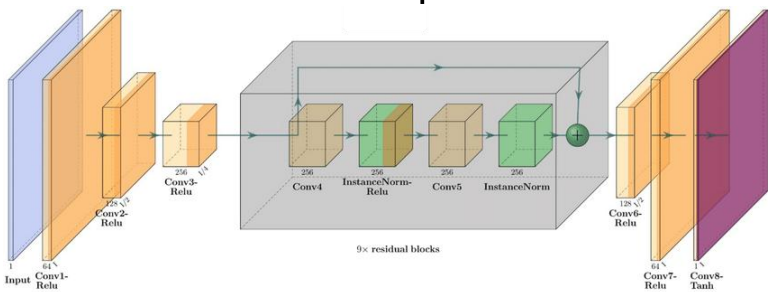
F





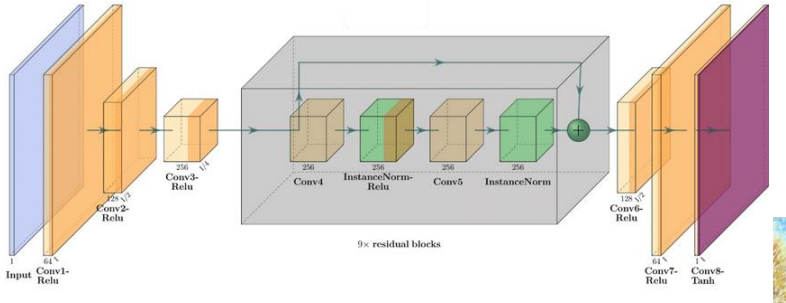
$$\underbrace{||F(G(x)) - x||_1}_{\text{consistencia del ciclo X→Y→X}} = \|\hat{x} - x\|_1 = \sum_i |\hat{x}_i - x_i|$$

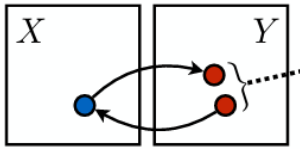
F



Ciclo2

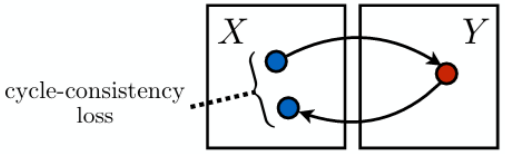
G





$$\underbrace{||G(F(y)) - y||_1}_{\text{consistencia del ciclo Y→X→Y}} \qquad \|\hat{y} - y\|_1 = \sum_i |\hat{y}_i - y_i|$$

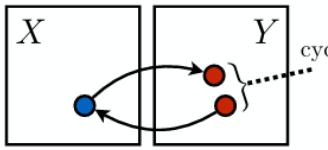
1. Los generadores están conectados en ciclo:



$$\underbrace{||F(G(x)) - x||_1}_{\text{consistencia del ciclo X→Y→X}} = \|\hat{x} - x\|_1 = \sum_i |\hat{x}_i - x_i|$$

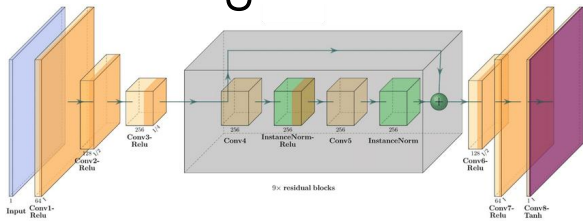
$$\mathcal{L}_{cyc} = \underbrace{||F(G(x)) - x||_1}_{\text{consistencia del ciclo X→Y→X}} + \underbrace{||G(F(y)) - y||_1}_{\text{consistencia del ciclo Y→X→Y}}$$

Se asegura que haya coherencia estructural en la traducción (contenido preservado).
Si no la incluimos notaríamos cambios arbitrarios de forma o disposición.



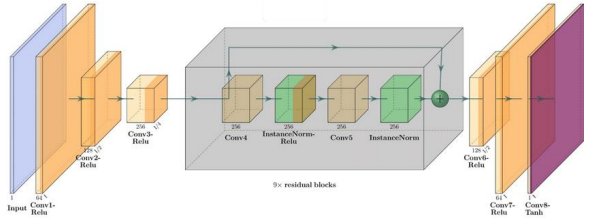
$$\underbrace{||G(F(y)) - y||_1}_{\text{consistencia del ciclo Y→X→Y}} \qquad \|\hat{y} - y\|_1 = \sum_i |\hat{y}_i - y_i|$$

G

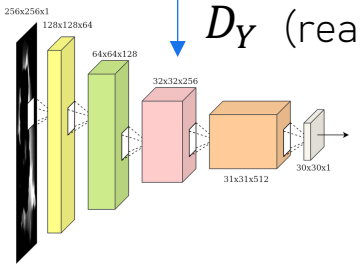


Ciclo 1

F

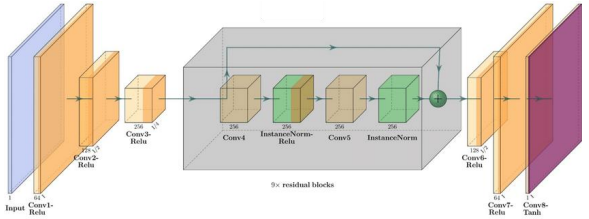


2. Cada generador está enfrentado a un discriminador (por la pérdida adversarial.



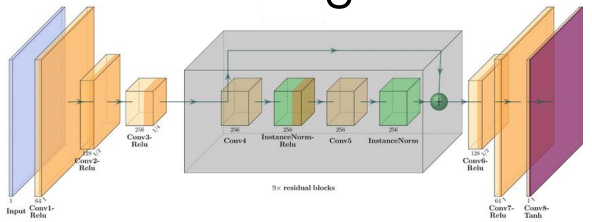
D_Y (real o falsa)

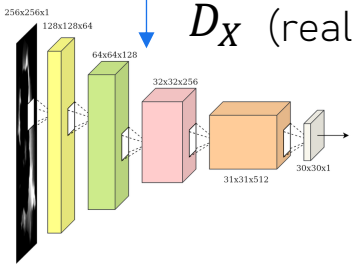
F



Ciclo2

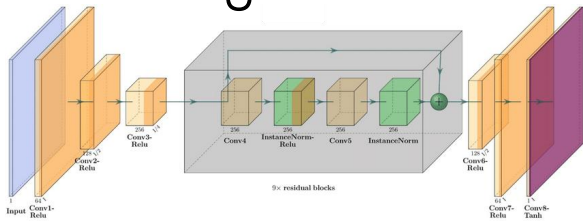
G





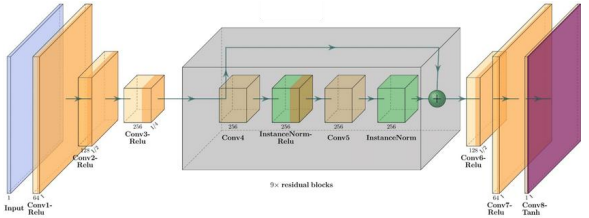
D_X (real o falsa)

G

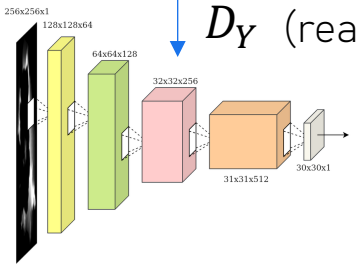


Ciclo 1

F



2. Cada generador está enfrentado a un discriminador (por la pérdida adversarial.

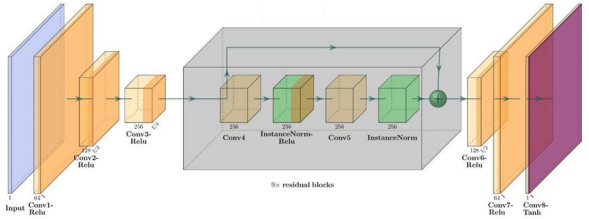


D_Y (real o falsa)

Promedio de errores locales es el que retroalimenta al generador.

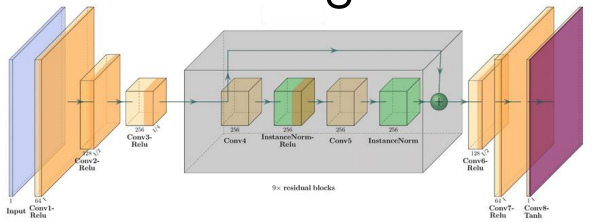
- Los *generadores cooperan entre sí* (vía la pérdida de ciclo), y
- *Compiten con sus discriminadores* (vía la pérdida adversarial).

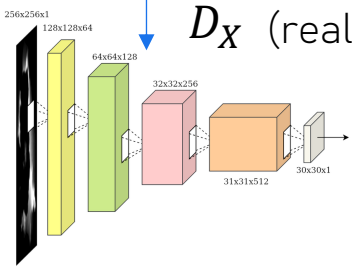
F



Ciclo2

G





D_X (real o falsa)

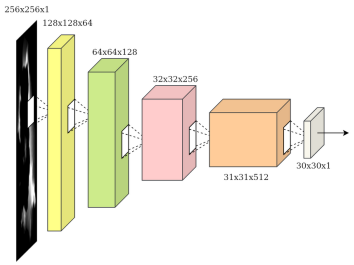
Relación	Mecanismo	Qué busca	Efecto
Competencia (G ↔ D)	Pérdida adversarial	G intenta engañar a D; D intenta detectar falsos	Mejora el realismo visual
Cooperación (G ↔ F)	Pérdida de ciclo	F debe revertir lo que hace G y viceversa	Preserva el contenido y coherencia
Equilibrio global	Suma ponderada de ambas pérdidas		Genera imágenes realistas y coherentes estructuralmente

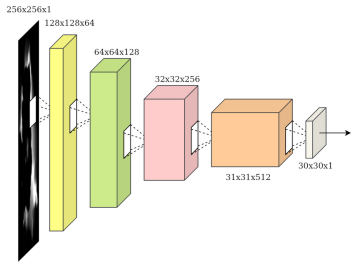
$$\mathcal{L}_{total}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda_{cyc}\,\mathcal{L}_{cyc}(G, F)$$

CycleGAN

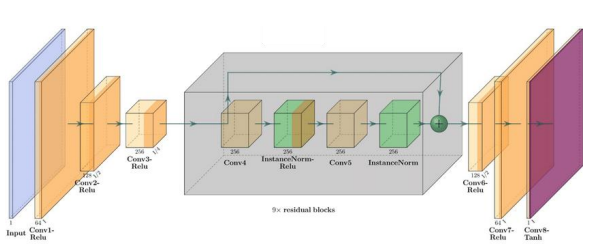
Entrenamiento

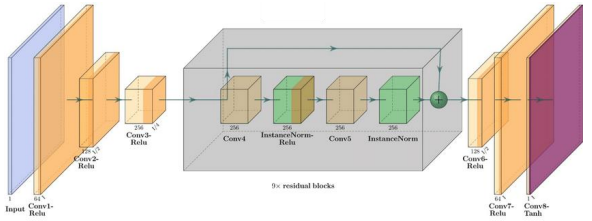
- El proceso alterna entre actualizar:
  1. Los **discriminadores** (maximizan la probabilidad de distinguir reales y falsas).





2. Los **generadores** (minimizan la pérdida adversarial + ciclo).





CycleGAN

Ventajas

- No requiere datasets pareados.
- Produce resultados visualmente coherentes entre dominios complejos.
- Mantiene la estructura general de la imagen (gracias a la pérdida de ciclo).

CycleGAN

Limitaciones

- Puede sufrir de modo colapso si las distribuciones son muy diferentes.
- No garantiza una correspondencia semántica perfecta.
- Entrenamiento costoso (cuatro redes en paralelo).

CycleGAN

Aplicaciones

- Conversión artística (Monet ↔ foto real).
  - CycleGAN puede aprender el estilo de un artista (por ejemplo, Monet o Van Gogh) y aplicarlo a fotografías reales, generando imágenes con la **paleta de colores, texturas y trazos** característicos del pintor.


- Mapeo médico (MRI ↔ CT).
  - En imágenes médicas, puede aprender a traducir entre modalidades como **resonancia magnética (MRI)** y **tomografía (CT)**, sin requerir estudios del mismo paciente en ambas modalidades.
  - Esto facilita generar imágenes complementarias o entrenar modelos diagnósticos cuando un tipo de estudio es escaso.

- Mejora de datos sintéticos ↔ reales.
  - Se utiliza para cerrar la brecha entre **datos simulados** y **datos reales**.
    - Ejemplo: convertir imágenes generadas por simuladores (sintéticas) en versiones más realistas para entrenar redes de visión por computadora, sin recopilar costosos datos reales.

CycleGAN

Aplicaciones

- Traducción de dominios en robótica o conducción autónoma.
  - Permite adaptar la percepción visual de un sistema entrenado en un entorno simulado a uno real (o viceversa).
  - Ejemplo: un coche autónomo entrenado en simulación puede usar CycleGAN para traducir imágenes del mundo real al "estilo" del simulador, facilitando la transferencia dominio-simulador ↔ real y mejorando la generalización.

Línea de tiempo de IA Generativa

1. La era de las VAEs y GANs (2013-2017)

StyleGAN

Introduce un nuevo enfoque en la arquitectura del generador al incorporar un "mecanismo de estilo" que permite un control más preciso y jerárquico sobre atributos visuales en la imagen generada. Este modelo logra un nivel sin precedentes de realismo y control sobre rasgos faciales y estructuras globales, marcando un hito en la generación de rostros sintéticos.

2019

·A partir de una sola imagen de una persona, puedes generar **versiones más jóvenes**, **con barba**, **con gafas** o **con diferente iluminación** sin necesidad de volver a entrenar la red.
·Retocar el **cabello**, el **maquillaje** o la **expresión facial** de una persona real conservando la coherencia visual.
·Puedes **mezclar estilos** de distintas imágenes: por ejemplo, la pose de una, la textura de piel de otra, y el peinado de una tercera.

Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019), 4401–4410.
https://arxiv.org/abs/1812.04948

Temario

1. Introducción.
2. Matemáticas esenciales para el aprendizaje profundo
3. Fundamentos de las redes neuronales profundas
4. Autocodificadores variacionales
5. Redes generativas adversarias (GANs)
6. Modelos autoregresivos
7. Modelos de normalización de flujo
8. Modelos basados en energía
9. Modelos de difusión