

# Aumento de la eficiencia de las redes generativas adversarias mediante el uso de transferencia de conocimiento

TITULACIÓN:  
Máster en Inteligencia Artificial

Curso académico:  
2020-2021

Lugar de residencia, mes y año:  
Barcelona, noviembre 2021

Alumno/a:  
Martí Isasi, Javier

D.N.I:  
26745202C

Director:  
Fuentes Hurtado, Félix José

Convocatoria:  
Tercera

Orientación:  
Investigación

Créditos:  
12 ECTS

*A mis padres.*

# Resumen

En los últimos años, las redes generativas basadas en el aprendizaje profundo están cobrado cada vez mayor interés debido a las mejoras sorprendentes en el campo. Estas redes han alcanzado una increíble capacidad para producir piezas de contenido muy realistas de diversos tipos, como imágenes, textos, sonidos y vídeos. Las redes generativas adversarias (GANs) son un tipo de redes generativas que se engloban dentro del campo de la visión artificial o visión por computador (*computer vision*). Permiten la generación de imágenes de gran calidad y resolución que incluso el ojo humano no puede distinguir si son reales o falsas.

Sin embargo, estos modelos necesitan normalmente de un conjunto de datos muy grande, algo que no siempre es posible, y que incluso cuando lo es, requiere grandes esfuerzos de tiempo y dinero. El entrenamiento con pocos datos suele provocar sobreajustes en el discriminador, lo que deriva en problemas de divergencia. Además, estos modelos requieren de un gran poder de computación que conlleva a largos tiempos de entrenamiento. Todo ello limita la aplicabilidad de este tipo de redes en tareas comunes del mundo real.

Con este trabajo se pretende aumentar la eficiencia de las redes generativas adversarias, tratando de conseguir los resultados deseados con muchos menos datos de entrenamiento y con un poder computacional notoriamente inferior.

Se estudia el impacto que supone la aplicación de la técnica de transferencia de conocimiento y ajuste fino. También se estudia la efectividad de otros métodos, en concreto, la técnica de aumento de datos y las técnicas de regularización. Se proponen nuevos diseños GAN que combinan estos métodos y se entrena en conjuntos de datos con un diferente número de muestras y variabilidad.

Se demuestra que, haciendo uso de la técnica de transferencia de conocimiento y ajuste fino, es posible obtener resultados de gran calidad utilizando un número de imágenes de entrenamiento increíblemente bajo. Además, se manifiesta un aumento de la eficiencia en los nuevos modelos diseñados que combinan las técnicas de transferencia de conocimiento y ajuste fino con las técnicas de aumento de datos y regularización. Los resultados obtenidos demuestran grandes mejoras de rendimiento en cuanto a la reducción de tiempo de entrenamiento, la estabilidad de los modelos y la calidad de las imágenes generadas.

En los entrenamientos de un máximo de 8K iteraciones en el conjunto de datos de 100 imágenes, se consigue mejorar el FID de 343.95 a 50.96 tras la aplicación de las técnicas de transferencia de conocimiento y ajuste fino. Además, se mejoran los resultados en un nuevo diseño que incluye la técnica de aumento de datos, alcanzando un FID de 37.90. Por otro lado, se consigue una mayor estabilidad del entrenamiento mediante el uso de técnicas de regularización de consistencia.

Todas estas mejoras amplían significativamente los horizontes de aplicabilidad de las redes generativas adversarias.

# Glosario

- Activaciones Maxout: Función que genera los elementos máximos de una distribución.
- Activaciones ReLU: Función que convierte los valores negativos de una distribución de datos a 0.
- Ajuste fino: Ajuste del entrenamiento de una red neuronal aprovechando el conocimiento del modelo de una red neuronal preentrenada.
- Aprendizaje automático: Tipo de inteligencia artificial que proporciona a las computadoras la capacidad de aprender, sin ser programadas explícitamente para ello.
- Aprendizaje no supervisado: Tipo de aprendizaje automático que estudia la estructura intrínseca de los datos.
- Aprendizaje profundo (*deep learning*): Subconjunto del aprendizaje automático basado en redes neuronales de varias capas de entrada, salida y ocultas.
- Aprendizaje supervisado: Tipo de aprendizaje automático que trata de deducir una función a partir de datos de entrenamiento.
- Aumento de datos (*data augmentation*): Técnica empleada en el entrenamiento de redes neuronales profundas que genera nuevas muestras de entrenamiento a partir de alteraciones de las originales sin afectar a las etiquetas de las clases.
- Autocodificador (*autoencoder*): Modelo generativo compuesto por un codificador y un decodificador cuyo objetivo es comprimir la distribución subyacente de los datos en un espacio latente de baja dimensión.
- *Big Generative Adversarial Network* (BigGAN): Arquitectura GAN diseñada por DeepMind en 2018 con extraordinarios resultados en generación de imágenes.

- Colapso del modo (*mode collapse*): Fallo común de entrenamiento de las GANs que se produce cuando el generador encuentra un pequeño número de muestras que engañan al discriminador.
- *Dataset*: Conjunto de datos, ya sea para el entrenamiento, validación o test de una red neuronal.
- Espacio latente: Entramado multidimensional abstracto de un modelo generativo. Contiene los valores de características de los datos de entrada extraídos por el modelo.
- *Fréchet Inception Distance* (FID): Métrica de evaluación de un modelo generativo que comprara la media y la covarianza de los datos sintéticos y reales en una red preentrenada.
- IA: Inteligencia Artificial.
- ImageNet: Base de datos de más de 14 millones de imágenes organizadas según la jerarquía de WordNet.
- *Inception Score* (IS): Métrica de evaluación de un modelo generativo que trata de puntuar la calidad de las imágenes sintéticas en función del realismo y la variedad.
- Inception-v3: Red neuronal convolucional preentrenada en el conjunto de datos ImageNet con 48 capas de profundidad que comenzó como un módulo para Googlenet.
- Minería de datos (*data mining*): Conjunto de procesos, métodos y técnicas que conducen a la extracción de conocimiento a partir de bases de datos.
- Modelo generativo (*generative model*): Arquitectura no supervisada dotada de algoritmos de aprendizaje profundo cuyo objetivo es aprender la distribución de los datos de entrada.
- Perceptrón multicapa (MLP): Red neuronal compuesta por múltiples capas usada para la clasificación de patrones linealmente separables.

- Ratio de aprendizaje (*learning rate*): Parámetro que marca la velocidad de aprendizaje de una red neuronal.
- Redes generativas adversarias (GAN): Modelo generativo compuesto por dos redes neuronales que compiten entre sí para ser más precisas en sus predicciones.
- Redes neuronales: Función de propósito general que puede resolver cualquier problema representable mediante ejemplos.
- Redes neuronales convolucionales o *Convolutional Neural Network* (CNN): Red neuronal ampliamente utilizado en el campo de la visión artificial donde el operador matemático básico es la convolución.
- Regularización de consistencia (consistency regularization): Técnica de aprendizaje semi-supervisado que pretende mejorar la estabilidad del entrenamiento de un modelo reduciendo la sensibilidad a la perturbación extra impuesta a las muestras de entrada.
- Retropropagación (*backpropagation*): Algoritmo de cálculo del gradiente utilizado para entrenar los pesos de una red neuronal.
- Tamaño de lote (*batch size*): Número de muestras de entrenamiento utilizados en una iteración del entrenamiento de un modelo.
- Técnicas de regularización: Técnicas empleadas en las redes neuronales profundas con el objetivo de mejorar la estabilidad del entrenamiento de los modelos.
- Transferencia de conocimiento: Técnica empleada en el entrenamiento de redes neuronales profundas que consiste en extraer las características de las redes preentrenadas y aplicarlas al modelo destino.
- Visión artificial o visión por computador (*computer vision*): Campo de la inteligencia artificial que se encarga de identificar y procesar entradas visuales replicando partes de la complejidad del sistema de visión.

# Índice

1. INTRODUCCIÓN .....	11
1.1. VISIÓN ARTIFICIAL .....	11
1.1.1. DEFINICIÓN DE LA VISIÓN ARTIFICIAL .....	11
1.1.2. EVOLUCIÓN DE LA VISIÓN ARTIFICIAL.....	12
1.1.3. APLICACIONES DE LA VISIÓN ARTIFICIAL.....	13
1.2. MODELOS GENERATIVOS .....	17
1.2.1. DEFINICIÓN DE LOS MODELOS GENERATIVOS.....	17
1.2.2. EVOLUCIÓN DE LOS MODELOS GENERATIVOS.....	18
1.3. REDES GENERATIVAS ADVERSARIAS .....	19
1.3.1. DEFINICIÓN DE LAS GANS.....	19
1.3.2. APLICACIONES DE LAS GANS .....	22
1.3.3. PROBLEMAS COMUNES DE LAS GANS.....	24
1.3.3.1. PODER DE COMPUTACIÓN Y NÚMERO DE MUESTRAS.....	25
1.3.3.2. NO CONVERGENCIA.....	25
1.3.3.3. COLAPSO DEL MODO.....	26
2. OBJETIVOS .....	27
3. ESTADO DEL ARTE.....	28
3.1. ARQUITECTURAS GANS .....	28
3.1.1. GENERATIVE ADVERSARIAL NETWORK (GAN), 2014.....	29
3.1.2. CONDITIONAL GENERATIVE ADVERSARIAL NETWORK (CGAN), 2014.....	29
3.1.3. DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK (DCGAN), 2015 .....	30
3.1.4. CYCLE-CONSISTENT GENERATIVE ADVERSARIAL NETWORK (CYCLEGAN), 2017 .....	31
3.1.5. PROGRESSIVE GROWING GENERATIVE ADVERSARIAL NETWORK (PROGAN), 2017.....	32
3.1.6. SELF-ATTENTION GENERATIVE ADVERSARIAL NETWORKS (SAGAN), 2018 .....	33
3.1.7. BIG GENERATIVE ADVERSARIAL NETWORK (BIGGAN), 2018 .....	34
3.1.8. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK (STYLEGAN), 2018 .....	34

3.1.9. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK 2 (STYLEGAN2), 2019 .....	35
3.1.10. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK 3 (STYLEGAN3), 2021 .....	37
3.2. TRANSFERENCIA DE CONOCIMIENTO Y AJUSTE FINO.....	37
3.2.1. TRANSFERRING GANS.....	40
3.2.2. BATCH STATISTICS ADAPTATION.....	40
3.2.3. MINEGAN.....	41
3.2.4. FREEZE-D.....	41
3.3. OTRAS TÉCNICAS DE AUMENTO DE EFICIENCIA DE LAS GANS .....	42
3.3.1. AUMENTO DE DATOS .....	42
3.3.1.1. DIFFERENTIAL AUGMENTATION (DIFFAUGMENT) .....	44
3.3.1.2. ADAPTIVE DISCRIMINATOR AUGMENTATION (ADA) .....	45
3.3.2. TÉCNICAS DE REGULARIZACIÓN .....	46
3.3.2.1. CONSISTENCY REGULARIZATION FOR GANS (CR) .....	46
3.3.2.2. A SIMPLE FRAMEWORK FOR CONTRASTIVE LEARNING OF VISUAL REPRESENTATIONS (SIMCLR) .....	47
4. MÉTODOS.....	48
4.1. METODOLOGÍA IMPLEMENTADA.....	48
4.2. ARQUITECTURA GAN.....	49
4.3. TRANSFERENCIA DE CONOCIMIENTO Y AJUSTE FINO.....	50
4.4. OTRAS TÉCNICAS DE AUMENTO DE EFICIENCIA DE LAS GANS .....	53
4.4.1. AUMENTO DE DATOS .....	53
4.4.2. REGULARIZACIÓN .....	56
5. RESULTADOS.....	58
5.1. LIMITACIONES DE HARDWARE.....	58
5.2. TECNOLOGÍAS EMPLEADAS.....	59
5.3. MÉTRICAS DE EVALUACIÓN.....	60
5.3.1. FUNCIONES DE PÉRDIDA .....	60
5.3.2. INCEPTION SCORE (IS) .....	60
5.3.3. FRÉCHET INCEPTION DISTANCE (FID) .....	61

5.4. CONJUNTOS DE DATOS EMPLEADOS .....	62
5.5. MODELOS DISEÑADOS .....	63
5.5.1. Modelo BigGAN (desde cero).....	63
5.5.2. Modelo BigGAN+TC.....	63
5.5.3. Modelo BigGAN+TC+AD .....	64
5.5.4. Modelo BigGAN+TC+AD+RC.....	66
5.6. EXPERIMENTOS Y RESULTADOS .....	67
6. CONCLUSIONES .....	73
7. LÍNEAS FUTURAS .....	76
8. BIBLIOGRAFÍA .....	79
9. ANEXOS.....	90
9.1. ANEXO 1. GRÁFICAS DE LAS FUNCIONES DE PÉRDIDAS DE LOS EXPERIMENTOS REALIZADOS	90
9.2. ANEXO 2. GRÁFICAS DEL FID DE LOS EXPERIMENTOS REALIZADOS .....	94

# 1. Introducción

En este apartado se analiza la situación del campo de la visión artificial y de los modelos generativos. Seguidamente, se estudian las redes generativas adversarias (GANs), detallando su arquitectura, aplicaciones y problemas comunes de entrenamiento.

## 1.1. VISIÓN ARTIFICIAL

### 1.1.1. DEFINICIÓN DE LA VISIÓN ARTIFICIAL

La visión artificial o visión por computador (*computer vision*) es la inteligencia artificial (IA) que se encarga de identificar y procesar imágenes, vídeos y otras entradas visuales replicando partes de la complejidad del sistema de visión. Si la IA permite a los ordenadores pensar, la visión artificial les permite ver, procesar, analizar y comprender las imágenes del mundo real.

La visión artificial se basa en una sólida comprensión y tratamiento de las entradas visuales tanto estáticas como en movimiento. Utiliza métodos estadísticos para desentrañar los datos utilizando modelos construidos con la ayuda de la geometría, la física y la teoría del aprendizaje. Permite extraer información numérica o simbólica que pueda ser interpretada y procesada por un ordenador.

### 1.1.2. EVOLUCIÓN DE LA VISIÓN ARTIFICIAL

Antes de la llegada del aprendizaje profundo, la visión por ordenador sólo funcionaba de forma limitada y requerían mucha codificación manual y esfuerzo por parte de desarrolladores y operadores humanos. La visión artificial se encuentra en un momento extraordinario de su desarrollo. El campo ha existido desde la década de 1950, pero sólo recientemente ha sido posible construir útiles sistemas capaces de superar a los humanos en algunas tareas relacionadas con la detección y etiquetado de objetos.

Este florecimiento ha sido impulsado gracias a las innovaciones en aprendizaje profundo, los avances en los recursos de *hardware*, la disponibilidad de computación en la nube y el acceso a una enorme cantidad de datos visuales (cada día se comparten más de 3000 millones de imágenes en Internet). Todo ello facilita la investigación y la resolución de problemas mediante métodos de visión por ordenador.

La llegada del aprendizaje automático ha proporcionado un enfoque diferente para resolver los problemas de visión por computador. Es entonces cuando se comienzan a utilizar algoritmos de aprendizaje estadístico como la regresión lineal, la regresión logística, los árboles de decisión o las máquinas de vectores de soporte (SVM) para la detección de patrones, la clasificación de imágenes y la detección de objetos. El aprendizaje automático ha ayudado a resolver muchos problemas que históricamente suponían un reto para las herramientas y enfoques clásicos de desarrollo de *software*.

El aprendizaje profundo está basado en las redes neuronales y proporciona un enfoque fundamentalmente diferente para realizar el aprendizaje automático. Al proporcionar a una red neuronal muchos ejemplos etiquetados de un tipo específico, será capaz de extraer patrones comunes y transformarlos en una expresión matemática que ayudará a clasificar futuras piezas de información. La

mayoría de las aplicaciones actuales de visión artificial, como la detección del cáncer, los coches autónomos y el reconocimiento facial, utilizan el aprendizaje profundo.

Los primeros experimentos en este campo se iniciaron en la década de 1960 y se utilizaron por primera vez en el ámbito comercial para distinguir entre texto escrito a mano y a máquina en los años 70.

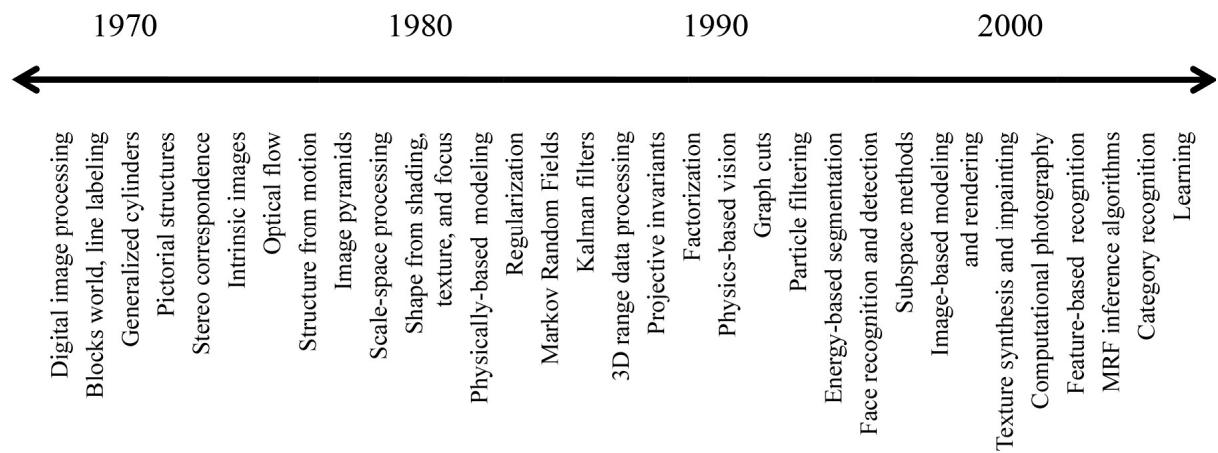


Figura 1.1: Una cronología aproximada de algunos de los temas más activos de la investigación en visión por ordenador [74].

### 1.1.3. APLICACIONES DE LA VISIÓN ARTIFICIAL

La visión artificial tiene una gran variedad de aplicaciones como son: la localización de objetos, la clasificación, la segmentación semántica, la segmentación de instancias, la localización dinámica, la generación sintética de imágenes, la verificación de objetos, los sistemas de recuperación de imágenes (CBIR) o la transferencia de estilo en imágenes y vídeo [90].

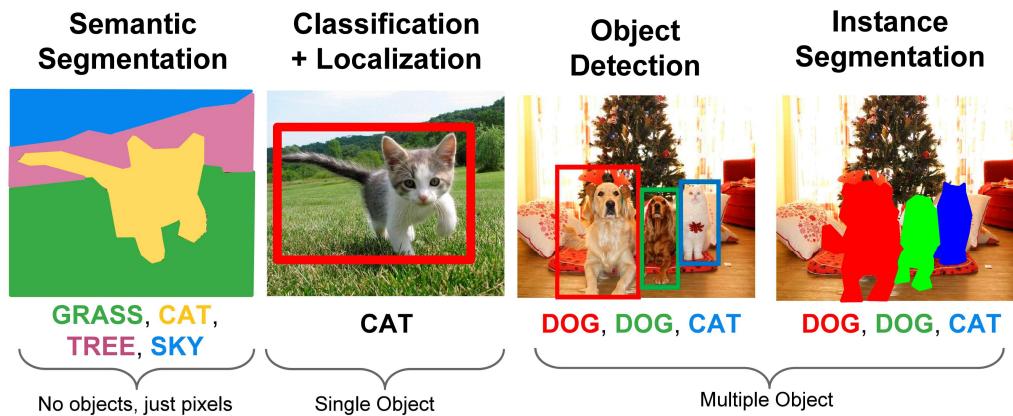


Figura 1.2: Diferentes aplicaciones en Visión Artificial [42].



Figura 1.3: Ejemplo de imágenes generadas por la arquitectura StyleGAN2 [36].

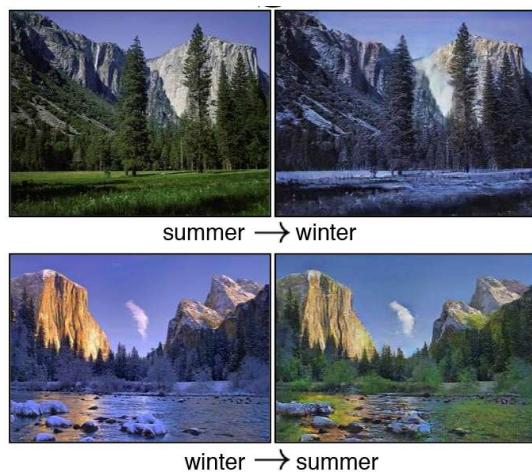


Figura 1.4: Método de transferencia de estilo mediante *Cycle-Consistent Adversarial Networks*, entrenado en fotos de invierno y verano de Yosemite [90].

Todas estas aplicaciones del mundo real demuestran la importancia de la visión artificial en los negocios, el entretenimiento, el transporte, la sanidad y la vida cotidiana. Un factor clave para el crecimiento de estas aplicaciones es la avalancha de información visual que fluye desde los teléfonos inteligentes, los sistemas de seguridad, las cámaras de tráfico y otros dispositivos con instrumentos visuales. Estos datos pueden desempeñar un papel importante en la industria 4.0 en sectores tan variados como la industria alimentaria, el comercio minorista, el transporte, la sanidad, el turismo, la agricultura, la ganadería, los procesos de fabricación, el control de calidad, la seguridad, la defensa, la industria del videojuego, la industria cinematográfica, la industria inmobiliaria, la industria publicitaria, el diseño, la moda, la medicina, la ingeniería, la arquitectura...

Algunas aplicaciones prácticas son: el reconocimiento óptico de caracteres (OCR) y de matrículas (ANPR) (figura 1.5a), la inspección de máquinas (figura 1.5b), el reconocimiento de objetos en la venta al por menor (figura 1.5c), la construcción de modelos 3D a partir de fotografías aéreas, el registro de imágenes preoperatorias e intraoperatorias (figura 1.5d), la detección de obstáculos inesperados en conducción (Figura 1.5e), la fusión de imágenes generadas por ordenador (CGI) con secuencias de acción real, la captura de movimiento, el análisis del tráfico en las carreteras (figura 1.5f), la vigilancia en aeropuertos o el reconocimiento de huellas dactilares y biometría para el control de accesos.

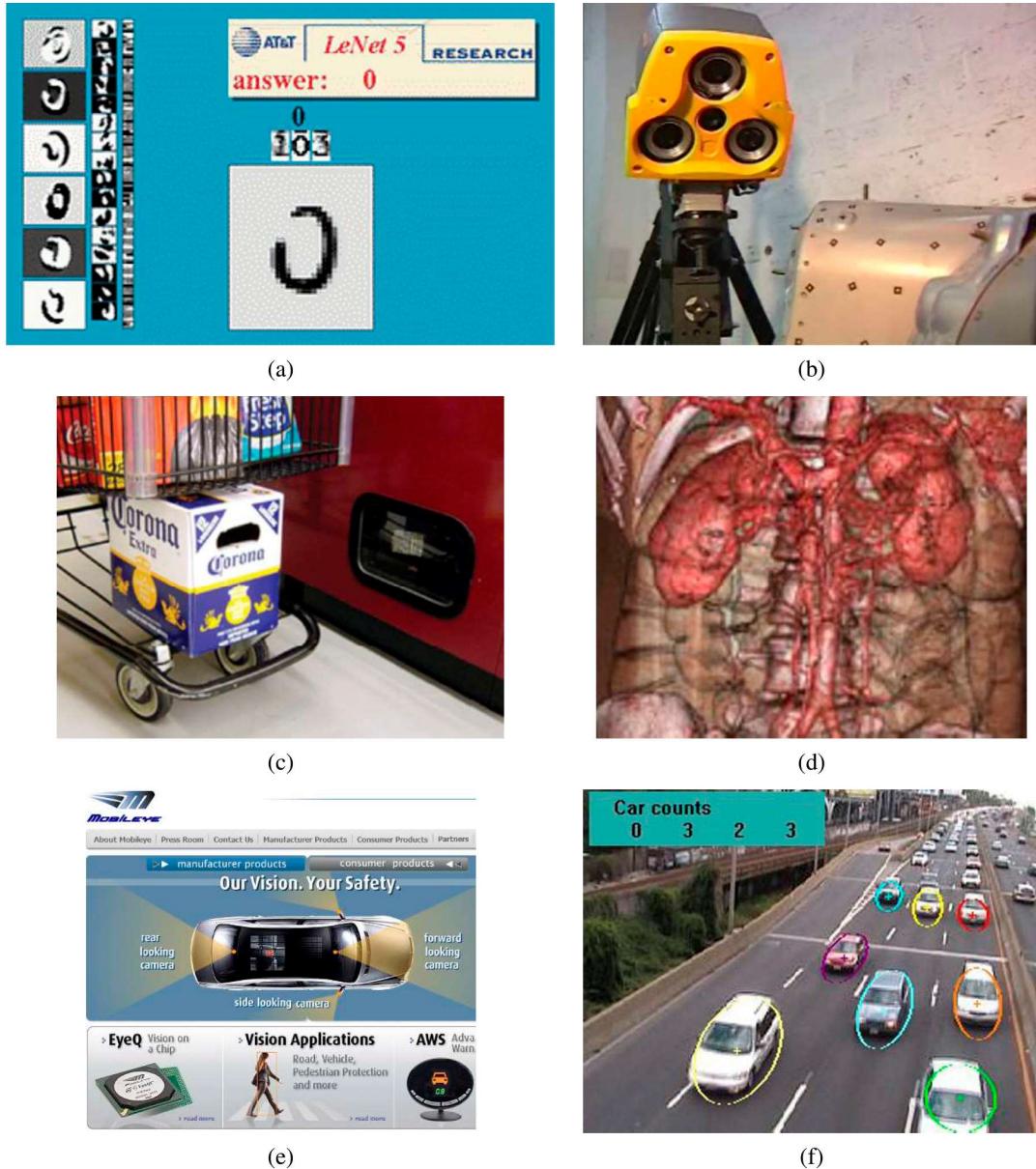


Figura 1.5: Ejemplos de aplicaciones industriales de visión artificial: (a) reconocimiento óptico de caracteres (OCR) (b) inspección mecánica (c) comercio minorista (d) imágenes médicas (e) seguridad en el automóvil (f) vigilancia y control del tráfico [74].

## 1.2. MODELOS GENERATIVOS

### 1.2.1. DEFINICIÓN DE LOS MODELOS GENERATIVOS

Los modelos generativos son una rama de la visión artificial que utiliza técnicas de aprendizaje no supervisado. Un modelo generativo describe cómo se genera un conjunto de datos en términos de un modelo probabilístico. Al tomar muestras de este modelo, se permiten generar nuevos datos.

La idea central de las redes generativas es capturar la distribución subyacente de los datos. Esta distribución no puede observarse directamente, sino que debe inferirse de forma aproximada a partir de los datos de entrenamiento. Para entrenar un modelo generativo es necesario una gran cantidad de datos de un dominio concreto. Estos datos de entrada son utilizados en el entrenamiento de un modelo para generar datos similares. La intuición que subyace a este enfoque sigue una famosa cita de Richard Feynman:

*"Lo que no puedo crear, no lo entiendo".*

En la figura 1.6 se muestra un resumen de un proceso típico de los modelos generativos.

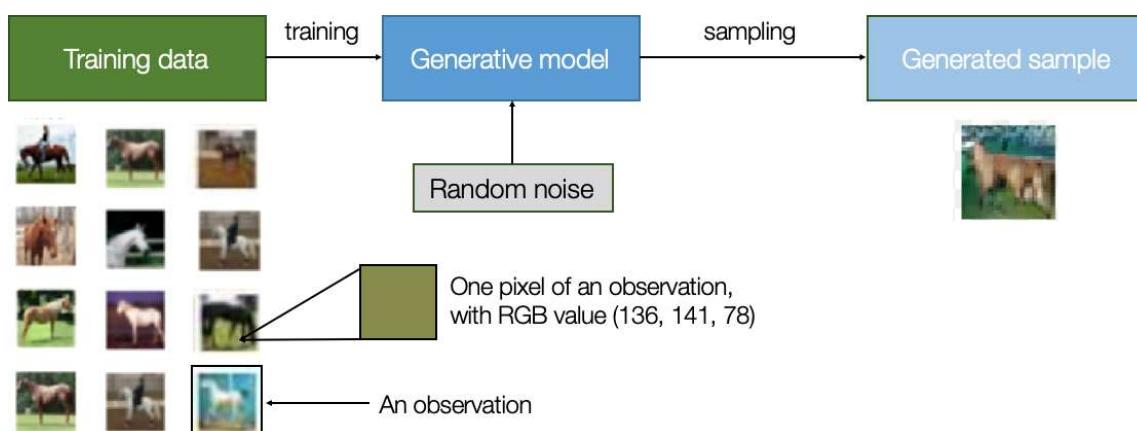


Figura 1.6: Proceso típico de los modelos generativos [22].

El objetivo que persiguen estas redes es la creación de nuevos conjuntos de características que parezcan haber sido generadas utilizando las mismas reglas que los datos originales. Conceptualmente, esta tarea es increíblemente difícil. Un modelo generativo debe ser probabilístico y no determinista para evitar la obtención del mismo resultado. Para ello, el modelo debe incluir un elemento estocástico (aleatorio) que influya en las muestras generadas por el modelo.

Las redes neuronales que se utilizan como modelos generativos tienen un número de parámetros significativamente inferior a la cantidad de entrenamiento, por lo que los modelos se ven obligados a descubrir e interiorizar eficazmente la esencia de los datos para poder generarlos.

### 1.2.2. EVOLUCIÓN DE LOS MODELOS GENERATIVOS

Los grandes avances conseguidos en el campo del aprendizaje profundo en los últimos años han impulsado el desarrollo de los modelos generativos.

Los primeros enfoques generativos profundos utilizaban autocodificadores (*autoencoders*). Fueron introducidos por primera vez en 1986 por Hinton y el grupo PDP [67]. Son un tipo de red neuronal no supervisada formada por un codificador y un decodificador que pretenden comprimir la distribución subyacente en un espacio latente  $z$  de baja dimensión, obligando a la red a aprender una representación compacta.

En 2013 Kingma et al. redefinieron el modelo original de los autocodificadores creando los autocodificadores variacionales (*variational autoencoders*) [38]. La principal diferencia con la configuración estándar es la modificación del espacio latente, que se divide en un vector de media  $\mu$  y un vector de desviación estándar  $\sigma$ .

En 2014 Ian Goodfellow presenta por primera vez las Redes Generativas Adversarias (GANs) [25]. Se componen de una red neuronal generadora y una red neuronal discriminadora y, a diferencia de los autocodificadores, crean espacios de altas dimensiones a partir de datos de más baja dimensión. Con el auge de la visión artificial y las redes neuronales convolucionales (CNN), las GANs han crecido rápidamente.

En el punto 1.3. se detalla la arquitectura GAN, sus aplicaciones y problemas comunes de entrenamiento. Mientras que en el punto 3.1. se describe el estado del arte de las GANs mediante una selección de los modelos más relevantes.

## 1.3. REDES GENERATIVAS ADVERSARIAS

### 1.3.1. DEFINICIÓN DE LAS GANS

Las redes generativas adversarias son modelos generativos que aplica técnicas de aprendizaje supervisado. Están compuestas por un submodelo generativo  $G$  y un submodelo discriminativo  $D$ . La red generativa captura la distribución de los datos y la red discriminativa estima la probabilidad de que una muestra provenga de los datos de entrenamiento y no de  $G$ . El procedimiento de entrenamiento de  $G$  consiste en maximizar la probabilidad de que  $D$  cometa un error. Los dos modelos se entrena juntos en un juego adversario hasta que el modelo discriminador es engañado aproximadamente la mitad de las veces, lo que significa que  $G$  está generando ejemplos plausibles. En el caso de que  $G$  y  $D$  estén definidos por perceptrones multicapa, todo el sistema puede ser entrenado con retropropagación.

El generador aprende una función que toma la distribución simple del ruido y la transforma en una distribución compleja que representa los datos deseados. Esta función transformadora es una función compleja donde los pesos de las redes son

los parámetros de la función. Una red neuronal profunda mapea la entrada al discriminador a un solo escalar permitiendo la probabilidad de que la entrada sea una muestra real.

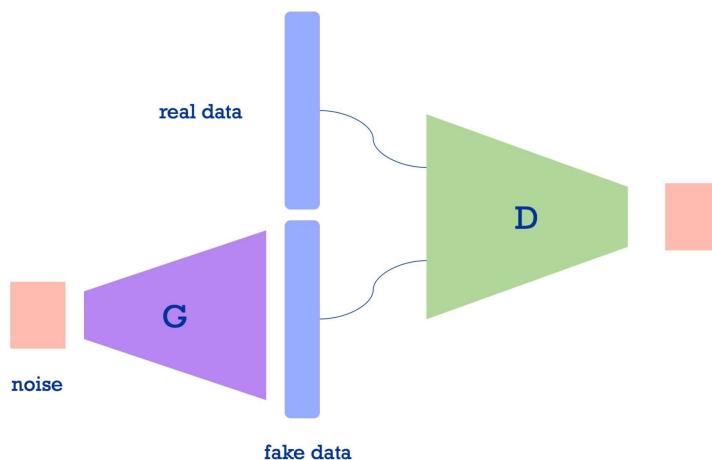


Figura 1.7: Representación esquemática de las GANs [89]. A la izquierda está el generador que aprende la transformación del ruido al dominio objetivo. A la derecha está el discriminador, que distingue esas muestras falsas generadas de las muestras reales.

Este proceso puede verse como un juego *min-max* de dos modelos. El estado de equilibrio en este juego hace que el generador produzca muestras falsas indistinguibles, y que el discriminador devuelva siempre 0.5, es decir, que adivine.

Para aprender la distribución  $p_g$  del generador sobre los datos  $x$ , se define una distribución simple de ruido  $p_z(z)$ , y luego se representa un mapeo al espacio de datos como  $G(z; \theta_g)$ , donde  $G$  es una función diferenciable representada por un perceptrón multicapa con parámetros  $\theta_g$ . También se define un perceptrón multicapa  $D(x; \theta_d)$  que produce un único escalar.  $D(x)$  representa la probabilidad de que  $x$  provenga de los datos  $y$  no de  $p_g$ . Se entrena  $D$  para maximizar la probabilidad de asignar la etiqueta correcta tanto a los ejemplos de

entrenamiento como a las muestras de  $G$ . Simultáneamente, se entrena  $G$  para minimizar  $\log(1 - D(G(z)))$ . Es decir,  $D$  y  $G$  juegan el siguiente juego *min-max* de dos jugadores con función de valor  $V(G, D)$ :

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

Siguiendo esta técnica, el generador intenta minimizar su error, y el discriminador se entrena para maximizar su precisión de clasificación.

Las GANs presentan ventajas y desventajas respecto a las redes generativas existentes hasta la fecha. Las desventajas son la inexistencia de una representación explícita de  $p_g(x)$  y la necesidad de una correcta sincronización de  $D$  con  $G$  durante el entrenamiento.  $G$  no debe entrenarse demasiado sin actualizar  $D$ , para evitar el "escenario Helvética" en el que  $G$  colapsa demasiados valores de  $z$  al mismo valor de  $x$  para tener suficiente diversidad para modelar  $p_{\text{data}}$ .

Una de las ventajas es que en las GANs no se actualiza la red generativa directamente con las muestras de entrada, sino sólo con los gradientes que fluyen a través del discriminador. Esto significa que los componentes de la entrada no se copian directamente en los parámetros del generador. Otra ventaja es que, a diferencia de los autocodificadores, las cadenas de Markov dejan de ser necesarias al no requerir de un muestreo del espacio latente. Sólo se utiliza la retropropagación en el entrenamiento para obtener gradientes mejorando computacionalmente el entrenamiento.

Las redes generativas adversarias son mucho más apropiadas para la generación sintéticas de imágenes que los autocodificadores, consiguiendo generar imágenes muy realistas.

### 1.3.2. APLICACIONES DE LAS GANS

Las aplicaciones de las redes generativas adversarias han experimentado un crecimiento asombroso en los últimos años.

Una aplicación de gran alcance consiste en ampliar conjuntos de datos pobres con imágenes sintéticas. Esta técnica puede ser utilizada en multitud de sectores como en procesos de fabricación para la detección de piezas defectuosas o en medicina para la detección de patologías [20].

Otras aplicaciones de las GANs incluyen: generación de imágenes de alta calidad y resolución, tareas de clasificación y regresión, traducción imagen-imagen [90], traducción texto-imagen [86, 63], superresolución [41], edición de imagen [61], predicción de vídeo [75], pintado de fotografías (*photo inpainting*) [59], registro de puntos en visión artificial y robótica [44] y generación de objetos en 3D [80]. Hay que alertar del uso fraudulento que se puede realizar mediante el uso de estas redes como la suplantación de identidad en el retoque de imágenes y vídeos. Esta es la técnica utilizada en los conocidos *Deep Fakes* [19].



Figura 1.8: Ejemplo de descripciones textuales y fotografías de aves generadas por *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*, 2016 [86].

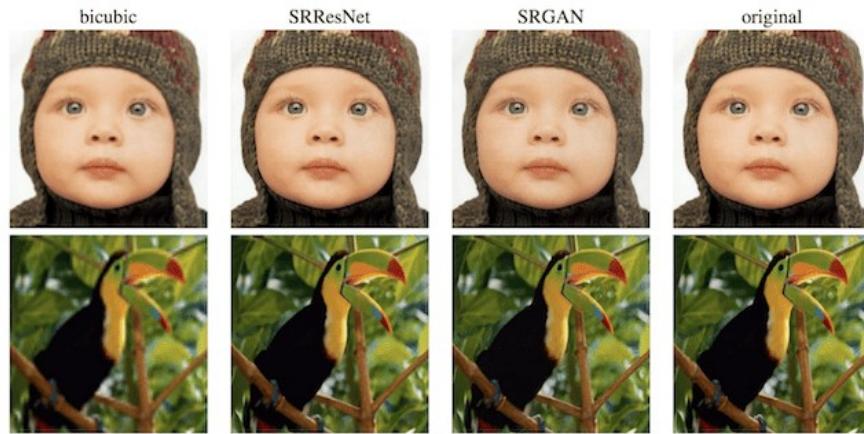


Figura 1.9: Ejemplo de imágenes generadas con superresolución mediante *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, 2016 [41].

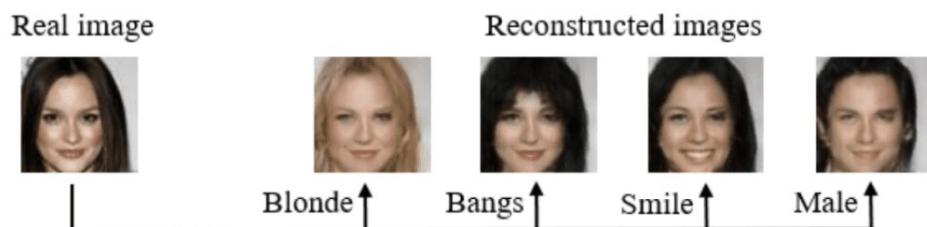


Figura 1.10: Ejemplo de edición de fotos de caras mediante *IcGAN: Invertible Conditional GANs For Image Editing*, 2016 [61].

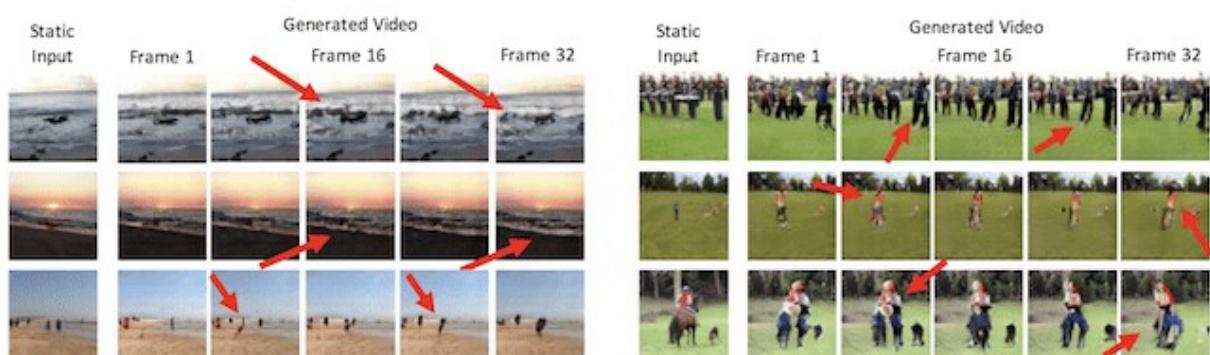


Figura 1.11: Ejemplo de fotogramas de vídeo generados a partir de la GAN definida en *Generating Videos with Scene Dynamics*, 2016 [75].

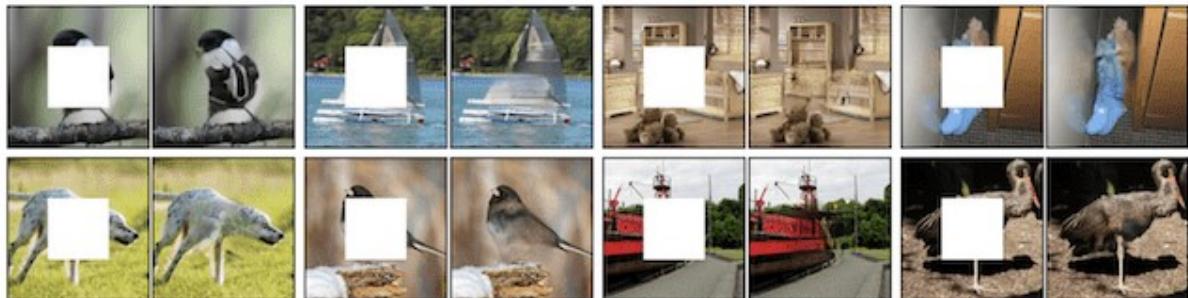


Figura 1.12: Ejemplo de *Inpainting* de fotografías generadas por GAN aplicando *Context Encoders: Feature Learning by Inpainting*, 2016 [59].

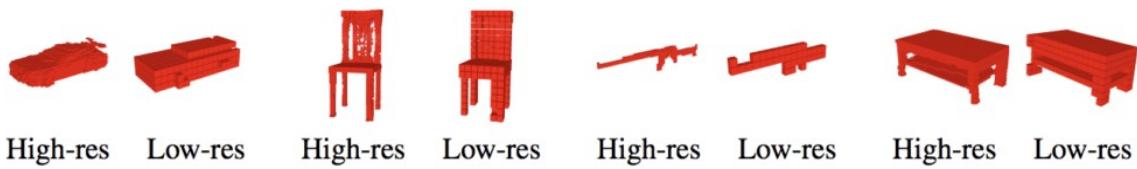


Figura 1.13: Ejemplo de objetos tridimensionales generados por la GAN definida en *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*, 2016 [80].

### 1.3.3. PROBLEMAS COMUNES DE LAS GANS

Es bien sabido que las GANs de alta calidad requieren una cantidad significativa de datos de entrenamiento y recursos computacionales. Además, presentan una serie de fallos comunes como la falta de convergencia. Estos problemas son líneas de investigación activa que han conseguido varios avances importantes en el funcionamiento del modelo GAN. Sin embargo, siguen existiendo algunos retos a solucionar.

A continuación, se detallan estos problemas comunes.

#### 1.3.3.1. PODER DE COMPUTACIÓN Y NÚMERO DE MUESTRAS

El entrenamiento de las GAN de alta calidad requiere una cantidad significativa de datos de entrada, poder de computación y tiempo de entrenamiento. Por ejemplo, las *Progressive GANs* [37] se entrena sobre 30K imágenes y requieren un mes de entrenamiento en una NVIDIA Tesla V100.

Conseguir reducir el número de muestras, el poder de computación y el tiempo de entrenamiento necesarios, es una cuestión crítica para ampliar los usos prácticos de estos modelos.

#### 1.3.3.2. NO CONVERGENCIA

Este problema sucede cuando el generador y el discriminador no logran alcanzar un equilibrio. Esto queda reflejado cuando la función de pérdida del generador y discriminador oscilan sin lograr una estabilidad. El entrenamiento de las GANs con muy pocos datos suele provocar un sobreajuste del discriminador, lo que hace que el entrenamiento sea divergente. En estos modelos es común que al comienzo del entrenamiento las funciones de pérdida oscilen. Pero a medida que transcurre el entrenamiento, el objetivo es que se logre una estabilidad. En problemas de no convergencia, la calidad de las muestras producidas por el generador no mejora.

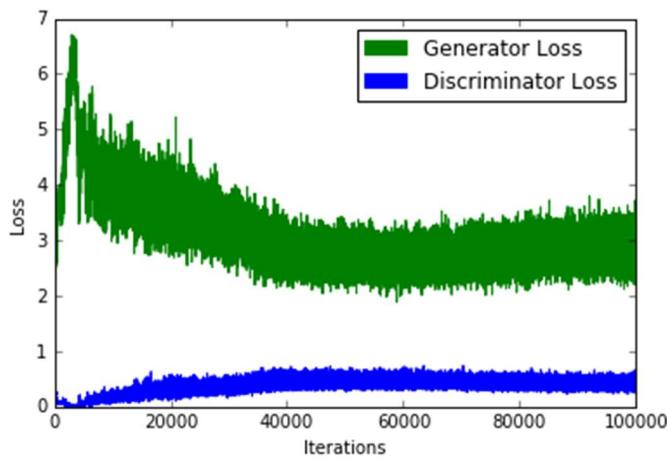


Figura 1.14: Gráfico de una función de pérdida de GAN [26].

### 1.3.3.3. COLAPSO DEL MODO

El colapso del modo (*mode collapse*) se produce cuando el generador encuentra un pequeño número de muestras que engañan al discriminador. Como resultado, los modelos no logran incrementar el número de imágenes. Los generadores tienden a encontrar una muestra que engaña al discriminador y pueden asignar cualquier punto del espacio latente a esta muestra. Esto significa que el gradiente de la función de pérdida se desploma a un valor cercano a cero. Incluso si el discriminador se entrena de nuevo para evitar ser engañado en un solo punto, el generador encontrará fácilmente otra muestra para engañar al discriminador. Esto se debe a que el generador es insensible a la entrada, y no está incitado a generar varias salidas.

## 2. Objetivos

A continuación, se enumeran los objetivos propuestos en el trabajo:

- Enmarcar la investigación dentro del campo de la visión artificial y de las redes generativas realizando un estudio sobre el campo.
- Entender la arquitectura de las redes generativas adversarias.
- Comprender los problemas comunes en el entrenamiento de las GANs.
- Realizar una investigación exhaustiva del estado del arte de las GANs.
- Estudiar el método de transferencia de conocimiento y ajuste fino.
- Elegir la arquitectura GAN base y entender su funcionamiento.
- Elegir los distintos conjuntos de datos y prepararlos para el entrenamiento.
- Diseñar un modelo que aumente la eficiencia de la red GAN elegida mediante el método de transferencia de conocimiento y ajuste fino.
- Investigar otros métodos que ayuden a mejorar la eficiencia de las GANs.
- Aplicar los métodos investigados en un nuevo modelo.
- Tratar de mejorar los resultados obtenidos optimizando la parametrización de los modelos diseñados.
- Evaluar los resultados obtenidos.
- Proponer futuras líneas de investigación.

## 3. Estado del arte

En este apartado se analiza el estado del arte de las arquitecturas de las redes generativas adversarias y de la técnica de transferencia de conocimiento y ajuste fino. Seguidamente, se detallará el estado del arte de otros métodos que mejoran de la eficiencia de las GANs. Estos son el aumento de datos y las técnicas de regularización.

### 3.1. ARQUITECTURAS GANS

Desde su primera aparición en 2014, y con el auge de la visión artificial y las redes neuronales convolucionales (CNN), las GANs han crecido rápidamente alcanzando unos resultados sorprendentes.



Figura 3.1: Evolución en la generación de rostros mediante modelos generativos [25][62][43][37][35][36].

A continuación, se detallan las principales arquitecturas GAN.

### 3.1.1. GENERATIVE ADVERSARIAL NETWORK (GAN), 2014

Como se detalló en el punto 1.2.2., la primera demostración empírica de la arquitectura de las redes generativas adversarias se describió en 2014 en el artículo de Ian Goodfellow, et al. titulado "*Generative Adversarial Networks*" [25]. La mayoría, si no todas, de las demás arquitecturas basadas en GANs están basadas en este modelo.

En el artículo se describe la arquitectura compuesta de un modelo generador que toma como entrada puntos de un espacio latente y genera una imagen, y un modelo discriminador que clasifica las imágenes como reales (del conjunto de datos) o falsas (producidas por el generador). Los modelos están compuestos por perceptrones multicapa (MLP) con activaciones *ReLU* en el generador y activaciones *Maxout* en el discriminador.

### 3.1.2. CONDITIONAL GENERATIVE ADVERSARIAL NETWORK (CGAN), 2014

Cinco meses después de la publicación de "*Generative Adversarial Networks*" [25], Mehdi Mirza y Simon Osindero introducen la "*Conditional generative adversarial network*" [51]. Se trata de una extensión de la arquitectura GAN que utiliza cierta información adicional a la imagen como entrada tanto para el generador como para el discriminador por medio de una capa adicional. Por ejemplo, si se dispone de etiquetas de clase, pueden utilizarse como entrada.

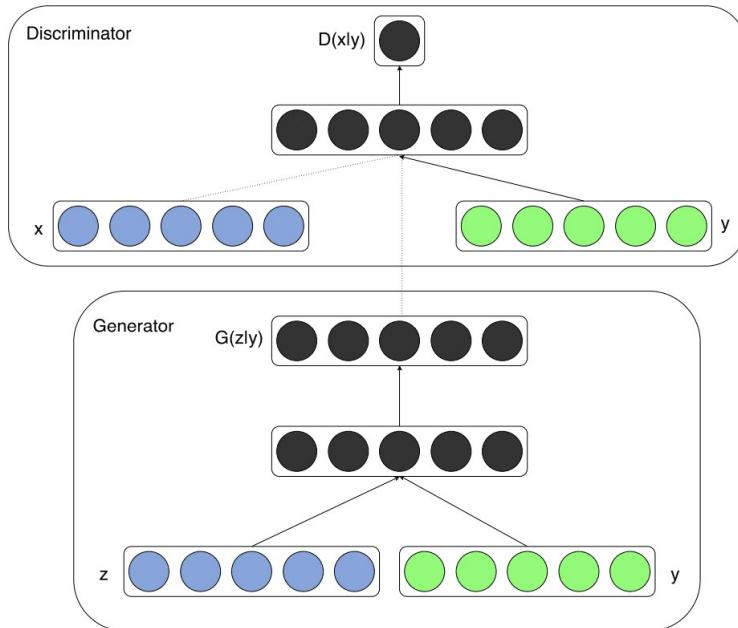


Figura 3.2: Ejemplo de la arquitectura del modelo de una *Red Adversarial Generativa Condicional* (*cGAN*) [51].

### 3.1.3. DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK (DCGAN), 2015

En 2015, Alec Radford, et al. publican el artículo "*Deep Convolutional Generative Adversarial Networks*" [62]. La red generativa adversaria profunda propuesta en este artículo es una extensión de la arquitectura GAN que utiliza redes neuronales convolucionales profundas tanto para los modelos generadores como para los discriminadores.

La publicación de DCGAN es de vital importancia para el desarrollo eficaz de modelos generadores de alta calidad y evidencia las restricciones del modelo GAN original. Esta arquitectura proporciona la base para el rápido desarrollo de un gran número de extensiones y aplicaciones de GAN.

### 3.1.4. CYCLE-CONSISTENT GENERATIVE ADVERSARIAL NETWORK (CYCLEGAN), 2017

En 2017, Jun-Yan Zhu, et al. publican el artículo "*Cycle-Consistent Adversarial Networks*" [90]. Este enfoque surgió originalmente para resolver el problema de la traducción de imagen a imagen, donde la muestra de entrada es de un dominio distinto a la de salida. Esta técnica mejora los enfoques anteriores que requerían conjuntos de datos exhaustivos y costosos de imágenes emparejadas.

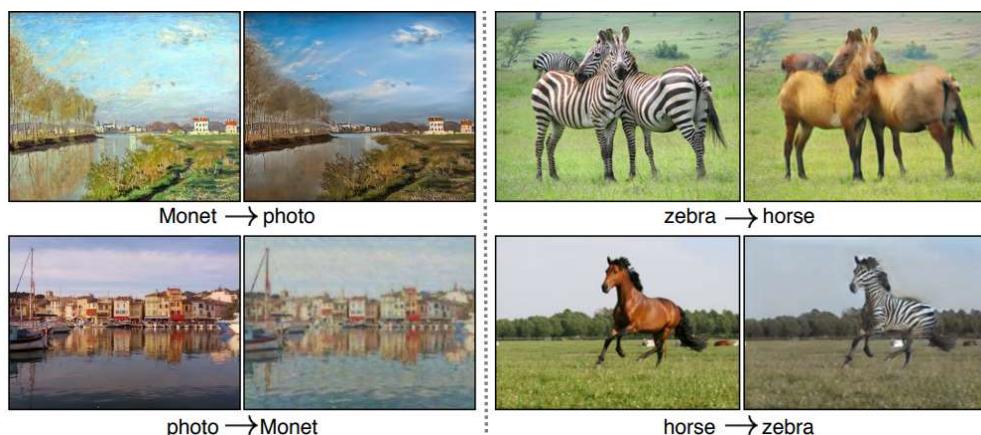


Figura 3.3: Ejemplos de traducción de imagen a imagen de CycleGAN [90].

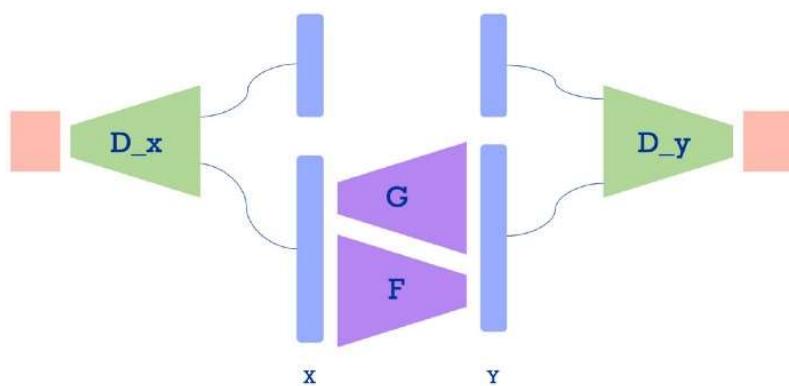


Figura 3.4: Representación esquemática de CycleGAN [90]. A la izquierda el discriminador  $D_x$  aprende a distinguir entre las muestras reales del dominio X y las muestras falsas generadas por  $F$ . A la derecha el discriminador  $D_y$  aprende a distinguir entre los datos reales del dominio Y y las muestras falsas generadas por  $G$ .

### 3.1.5. PROGRESSIVE GROWING GENERATIVE ADVERSARIAL NETWORK (PROGAN), 2017

En 2017, NVIDIA Labs (Terro Karras, et al.) publican el artículo "*Progressive Growing of GANs*" [37]. Pretende mejorar tanto la velocidad como la estabilidad del entrenamiento de las GAN. Exponen un cambio en la arquitectura y el entrenamiento de los modelos GAN que implica el aumento progresivo de la profundidad del modelo durante el proceso de entrenamiento.

Esto lo consiguen manteniendo el generador y el discriminador simétricos en profundidad durante el entrenamiento y añadiendo capas progresivamente. El documento sugiere entrenar primero el generador y el discriminador en imágenes de baja resolución y, posteriormente, añadir capas de forma gradual a lo largo del proceso de entrenamiento para aumentar la resolución. El modelo alcanzó excelentes resultados.

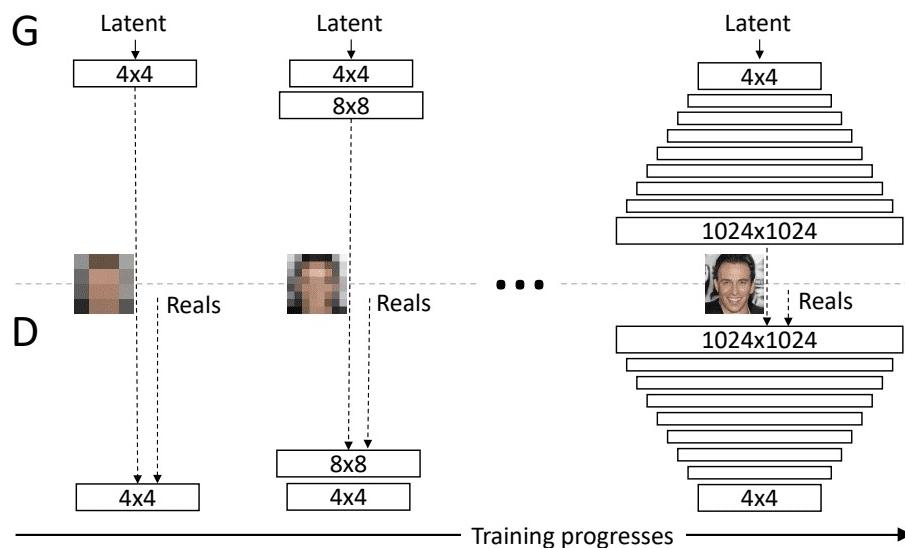


Figura 3.5: Ejemplo de entrenamiento de Progressive GAN [37].

### 3.1.6. SELF-ATTENTION GENERATIVE ADVERSARIAL NETWORKS (SAGAN), 2018

En 2018, Han Zhang, et al. publican el artículo "*Self-Attention Generative Adversarial Networks*" [84]. Las redes adversariales convolucionales tradicionales generan detalles de alta resolución en función de puntos espacialmente locales en mapas de características de baja resolución. Los autores proponen un modelo que genera imágenes aprovechando características complementarias en partes distantes de la imagen para generar objetos consistentes.

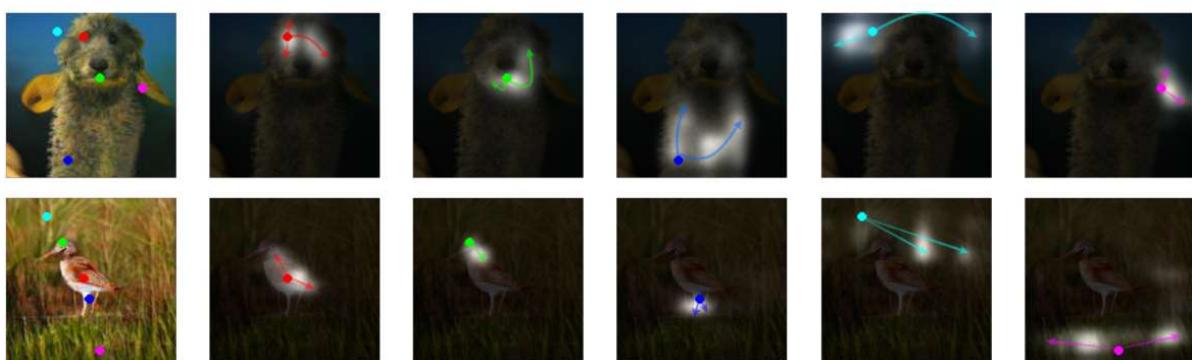


Figura 3.6: Ejemplo de funcionamiento de los mapas de atención de SAGAN. En cada fila, la primera imagen muestra cinco imágenes de consulta representativas con puntos codificados por colores. Las otras cinco imágenes son mapas de atención para esas imágenes de consulta, con las correspondientes flechas codificadas por colores que resumen las regiones más atendidas [84].

Este modelo fue considerado en el momento de la publicación como la GAN más avanzado, pero no por mucho tiempo.

### 3.1.7. BIG GENERATIVE ADVERSARIAL NETWORK (BIGGAN), 2018

En 2018, DeepMind (Andrew Brock, et al.) publica el artículo "*Large Scale GAN Training for High Fidelity Natural Image Synthesis*" [11]. La red BigGAN es un enfoque que demuestra cómo se pueden crear imágenes de salida de alta calidad ampliando las ideas del documento SAGAN [84]. La arquitectura del modelo se basa en una colección de mejores prácticas a través de una amplia gama de extensiones y modelos GAN.

BigGAN es capaz de generar imágenes de gran calidad en una amplia gama de clases de objetos, obteniendo extraordinarios resultados en el conjunto de datos ImageNet.

### 3.1.8. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK (STYLEGAN), 2018

En 2018, NVIDIA Labs (Tero Karras, et al.) publican el artículo "*The Style Generative Adversarial Network*" [35]. La red StyleGAN es una extensión de la arquitectura GAN que propone grandes cambios en el modelo generador. Incluye el uso de una red para mapear puntos del espacio latente a un espacio latente intermedio utilizado para controlar el estilo en cada punto del modelo generador. Por otro lado, introduce ruido como fuente de variación en cada punto del modelo generador.

El modelo resultante, además de generar imágenes de gran calidad, ofrece un control sobre el estilo de la imagen generada en diferentes niveles de detalle mediante la variación de los vectores de estilo y del ruido.

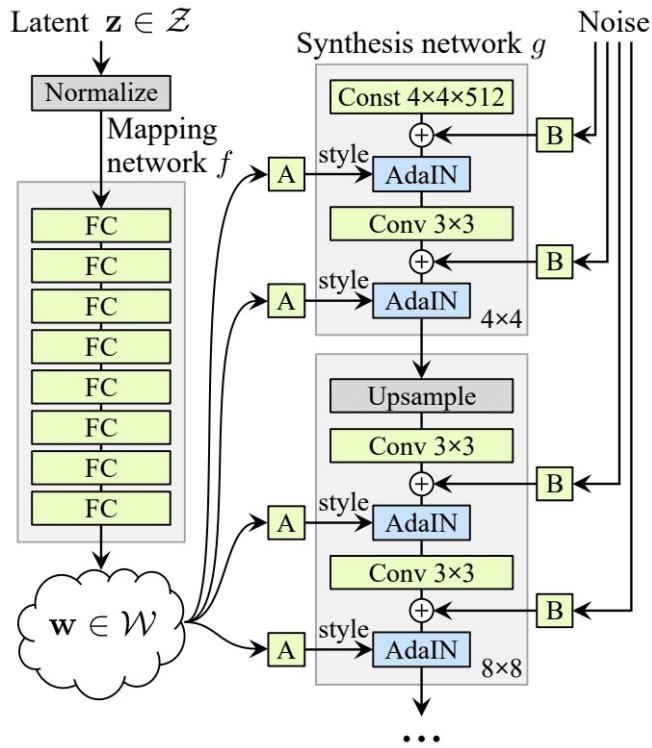


Figura 3.7: Resumen de la arquitectura del generador de StyleGAN [35].

### 3.1.9. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK 2 (STYLEGAN2), 2019

En 2019, NVIDIA Labs (Tero Karras, et al.) publican el artículo "*Analyzing and Improving the Image Quality of StyleGAN*" [36]. En este artículo se expone y analiza la red StyleGAN y propone cambios en la arquitectura del modelo y en los métodos de entrenamiento. Se rediseña la normalización del generador y se regulariza el generador para fomentar un buen condicionamiento en el mapeo del espacio latente a las imágenes.

Inspirado en el MSG-GAN, StyleGAN2 aplica el crecimiento progresivo haciendo uso de múltiples escalas de generación de imágenes sin requerir explícitamente

que el modelo lo haga. Se consigue a través de un aumento de la resolución de las muestras y la suma de las contribuciones de las salidas RGB correspondientes a las diferentes resoluciones. En el discriminador, se proporciona de forma similar las imágenes a cada bloque de resolución del discriminador.

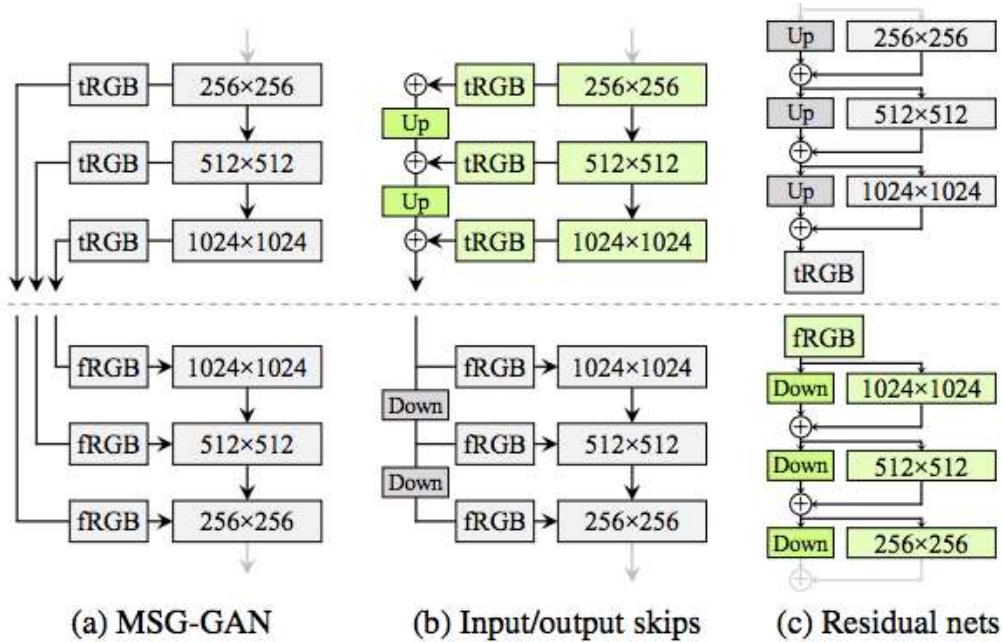


Figura 3.8: Ejemplos de tres arquitecturas de StyleGAN2 [36].

Además de mejorar la calidad de la imagen, el modelo tiene la ventaja adicional de que el generador es mucho más fácil de invertir. Esto permite atribuir de forma fiable una imagen generada a una red concreta. El modelo mejorado consigue grandes resultados en el modelado de imágenes, tanto en términos de las métricas de calidad de distribución existentes como de la calidad de imagen percibida.

### 3.1.10. STYLE-BASED GENERATIVE ADVERSARIAL NETWORK 3 (STYLEGAN3), 2021

En 2021, NVIDIA Labs (Terro Karras, et al.) publica el artículo "*Alias-Free Generative Adversarial Networks*" [33]. Este modelo pone en evidencia el problema del *aliasing* en la red de generadores que conlleva a una incorrecta representación de los detalles.

El modelo mejorado interpreta todas las señales de la red como continuas y realiza pequeños cambios que garantizan que la información no deseada no pueda filtrarse. Las redes resultantes coinciden con el FID de StyleGAN2. Sin embargo, difieren drásticamente en sus representaciones internas y son totalmente equivalentes a la translación y la rotación, incluso a escalas subpixelares. Este modelo allana el camino hacia modelos generativos más adecuados para el vídeo y la animación.

## 3.2. TRANSFERENCIA DE CONOCIMIENTO Y AJUSTE FINO

Una de las características atractivas de las redes neuronales profundas es su capacidad para transferir los conocimientos obtenidos de un dominio a otros dominios relacionados. Como resultado, las redes preentrenadas de alta calidad pueden ser entrenadas en dominios con relativamente pocos datos de entrenamiento. Dado el enorme esfuerzo que suele requerir el entrenamiento de las GANs, tanto a nivel computacional como de recopilación de datos, la reutilización de las GANs preentrenadas es un objetivo deseable.

La aplicación del ajuste fino (*fine tuning*) es una técnica extremadamente poderosa, ya que permite evitar entrenar toda una red desde cero. En su lugar,

se puede aprovechar arquitecturas de redes preexistentes, como los modelos de última generación entrenados en el conjunto de datos ImageNet. Permite iniciar el aprendizaje utilizando estos filtros mediante la transferencia de conocimiento. Se obtiene un modelo de aprendizaje de mayor precisión y menor esfuerzo que el entrenamiento desde cero.

Se emplean modelos de aprendizaje profundo que ya han sido entrenados en un conjunto de datos determinado. Normalmente, estas redes son arquitecturas de última generación como VGG, ResNet o Inception que han sido entrenadas en el conjunto de datos ImageNet. Contienen filtros ricos y discriminativos que pueden utilizarse en conjuntos de datos y etiquetas fuera de los que ya han sido entrenadas.

Mediante la técnica de transferencia de conocimiento se extraen las características de las redes preentrenadas al modelo destino y mediante el método de ajuste fino se modifica la arquitectura de las redes para poder reentrenar partes de su red.

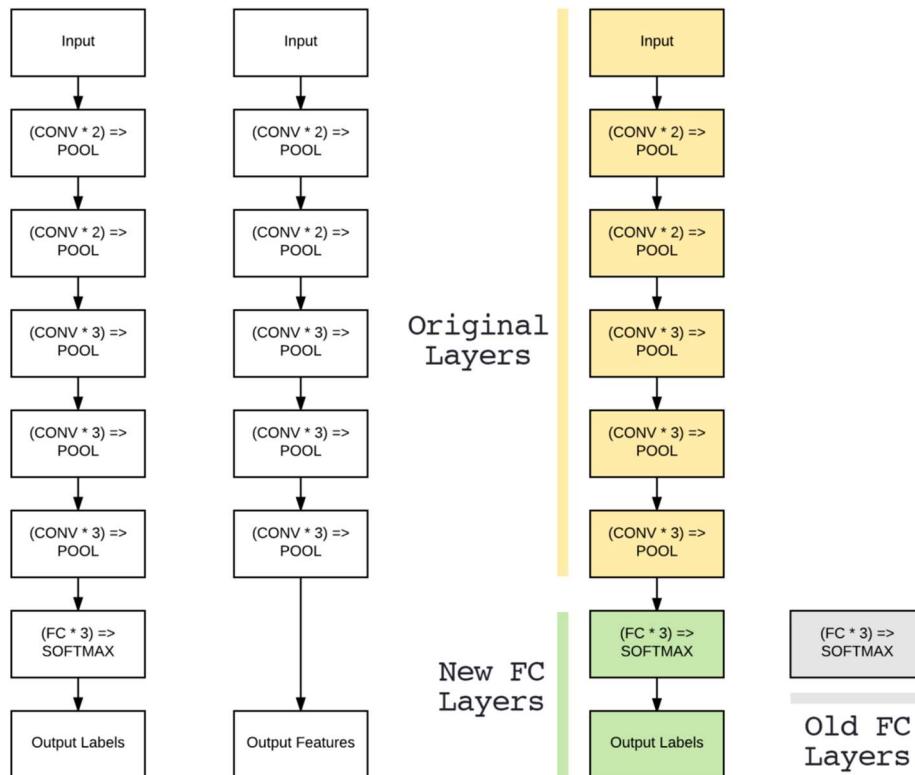


Figura 3.9: Ejemplo de ajuste fino mediante la trasferencia de conocimiento de la red preentrenada VGG16 [64]. Izquierda: La arquitectura original de la red VGG16. Centro: Eliminación de las capas FC de VGG16 y tratamiento de la capa final POOL como un extractor de características. Derecha: Eliminación de las capas FC originales y sustitución por nuevas capas FC. Estas nuevas capas de FC pueden ser ajustadas para el conjunto de datos específico (las antiguas capas FC ya no se utilizan).

A la izquierda de la figura 3.9 se representan las capas de la red VGG16. El conjunto final de capas (es decir, la "cabeza") corresponden a las capas totalmente conectadas junto con el clasificador *softmax*. Al realizar el ajuste fino, se elimina la cabeza de la red, al igual que en la extracción de características (centro). Sin embargo, cuando realizamos el ajuste fino, construimos una nueva cabeza totalmente conectada y la colocamos sobre la arquitectura original (derecha).

Mientras que la transferencia de conocimiento ha sido ampliamente estudiada para modelos discriminativos en visión por ordenador [52, 28, 56, 55] sólo unos pocos trabajos han explorado la transferencia de conocimiento para modelos generativos. A continuación, se detallan algunos de ellos.

### 3.2.1. TRANSFERRING GANS

En 2018 Yaxing Wang et al. publican un método de transferencia de conocimiento en su artículo titulado "*Transferring GANs: generating images from limited data*" [76]. Este artículo aplica transferencia de conocimiento a la red generadora consiguiendo acortar el tiempo de convergencia y mejorar la calidad de las imágenes generadas, especialmente cuando el número de muestras es limitado. Para ello, se estudia el impacto al dominio de destino, la inicialización de las GANs condicionales y la distancia relativa entre el dominio de origen y destino.

### 3.2.2. BATCH STATISTICS ADAPTATION

En 2018 Atsuhiro Noguchi et al. publican un método de transferencia de conocimiento en su artículo titulado "*Image Generation From Small Datasets via Batch Statistics Adaptation*" [54]. Esta técnica transfiere eficazmente el conocimiento de una red preentrenada en un dominio distinto al de destino. Se consigue un entrenamiento estable del generador y unas imágenes de mayor calidad gracias a la diversidad de los filtros aplicados en el generador.

### 3.2.3. MINEGAN

En 2019 Yaxing Wang et al. publican un método de transferencia de conocimiento en su artículo titulado "*MineGAN: effective knowledge transfer from GANs to target domains with few images*" [78].

En el artículo se propone un novedoso método de transferencia de conocimiento basado en la extracción del conocimiento más beneficioso para un dominio específico, ya sea a partir de una o varias GAN preentrenadas. Para ello, se utiliza una red minera que identifica qué parte de la distribución generativa de cada GAN preentrenada produce las muestras más cercanas al dominio objetivo. La red minera dirige eficazmente el muestreo hacia las regiones adecuadas del espacio latente, lo que facilita el ajuste posterior y evita las patologías de otros métodos, como el colapso del modo y la falta de flexibilidad.

El método consigue trasferir de forma satisfactoria el conocimiento a dominios con pocas imágenes, superando a los métodos existentes en la publicación del artículo.

### 3.2.4. FREEZE-D

En 2020 Sangwoo Mo et al. publican un método de transferencia de conocimiento en su artículo titulado "*Freeze the Discriminator: a Simple Baseline for Fine-Tuning GANs*" [52]. En este artículo se propone un simple ajuste fino que consiste en congelar las capas inferiores del discriminador obteniendo unos modelos sorprendentemente efectivos. Los autores implementaron este método sobre las arquitecturas StyleGAN [35] y SNGAN [50].

### **3.3. OTRAS TÉCNICAS DE AUMENTO DE EFICIENCIA DE LAS GANS**

Además de la técnica de transferencia de conocimiento y ajuste fino, existen otras técnicas muy poderosas que permiten aumentar la eficiencia de las GANs como la técnica del aumento de datos y la regularización de consistencia. A continuación, se detallan cada una de ellas.

#### **3.3.1. AUMENTO DE DATOS**

La técnica de aumento de datos (*data augmentation*) abarca una amplia gama de métodos utilizados para generar nuevas muestras de entrenamiento a partir de las originales, aplicando alteraciones y perturbaciones aleatorias, de modo que las etiquetas de las clases no se vean afectadas.

Al aplicar esta técnica, la red recibe constantemente nuevas versiones ligeramente modificadas de los datos de entrada, consiguiendo aprender características más robustas. De esta forma, se pretende paliar el sobreajuste típico en modelos estrenados con datos limitados aumentando la generalizabilidad del modelo.

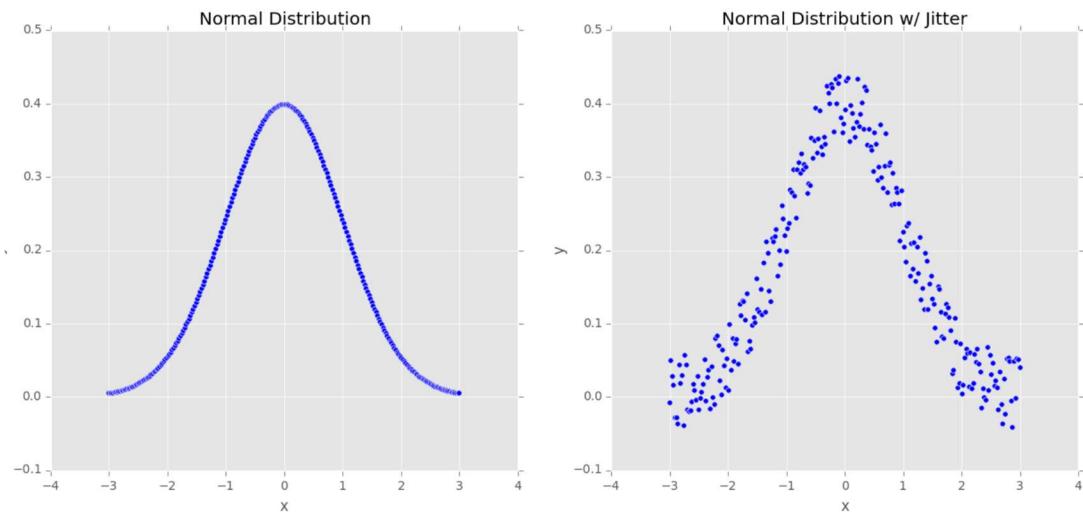


Figura 3.10: (izquierda) Muestra de 250 puntos de datos que siguen exactamente una distribución normal. (derecha) Distribución con una pequeña cantidad de fluctuaciones aleatorias. Este tipo de aumento de datos puede aumentar la capacidad de generalización de la red [55].

La figura 3.10 (izquierda) corresponde a una distribución normal con media cero y varianza unitaria. En la figura de la derecha se añaden algunos valores extraídos de una distribución aleatoria, más próxima a datos de imágenes del mundo real. La distribución resultante sigue siguiendo aproximadamente normal, pero no es una distribución perfecta como la de la izquierda. El entrenamiento de un modelo con estos datos alterados tiene más probabilidades de generalizar y evitar problemas de sobreajuste.

En el contexto de la visión por ordenador, el aumento de datos se presta de forma natural. Por ejemplo, se pueden obtener datos de entrenamiento adicionales a partir de las imágenes originales aplicando simples transformaciones geométricas aleatorias como traslaciones, rotaciones, cambios de escala, aplicación de ruido, modificaciones en el color y simetrías.

El rendimiento de las redes generativas adversarias se deteriora mucho con una cantidad limitada de datos de entrenamiento. Esto se debe principalmente a que el discriminador está memorizando el conjunto de entrenamiento. Para combatir este problema, varias técnicas basadas en el aumento de datos han demostrado mejorar la eficiencia de estas redes.

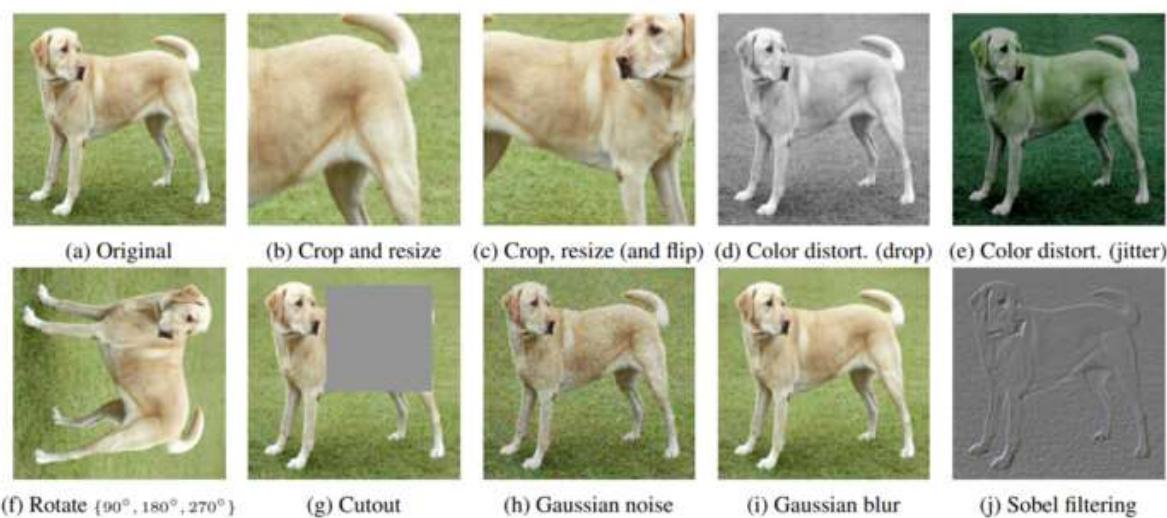


Figura 3.11: Técnica de aumento de datos utilizado en SimClr [16].

A continuación, se detallan varias técnicas que aplican la técnica de aumento de datos.

### 3.3.1.1. DIFFERENTIAL AUGMENTATION (DIFFAUGMENT)

En 2020 Shengyu Zhao et al. publican un método de aumento de datos en su artículo titulado "*Differentiable Augmentation for Data-Efficient GAN Training*" [54].

Se trata de un método sencillo que mejora la eficiencia de las GANs imponiendo varios tipos de aumentos diferenciables tanto en las muestras reales como en las falsas. En otros intentos del estado del arte anterior, se aplica un aumento de las imágenes reales produciendo un beneficio limitado. DiffAugment consigue un aumento diferenciable de las muestras, tanto las reales como las sintéticas, y una estabilización eficaz del entrenamiento que conduce a una mejor convergencia.

Consigue generar imágenes de alta calidad utilizando sólo 100 imágenes sin preentrenamiento, consiguiendo un resultado similar a los algoritmos de aprendizaje de transferencia de conocimiento existentes.

### 3.3.1.2. ADAPTIVE DISCRIMINATOR AUGMENTATION (ADA)

En 2020 Tero Karras et al. publican un método de aumento de datos en su artículo titulado "*Training Generative Adversarial Networks with Limited Data*" [34].

Este método realiza en esencia las mismas técnicas de aumento de datos que DiffAugment [54], basado en realizar un aumento diferenciable tanto en el discriminador como en el generador. A diferencia de DiffAugment, ADA está basado en la estrategia de aumento adaptativo.

Esta técnica considera la posibilidad de ajustar la fuerza de aumento en función del conjunto de datos o de la evolución del entrenamiento. Según los autores, estabiliza significativamente el entrenamiento de modelos con una disponibilidad de datos limitados.

### 3.3.2. TÉCNICAS DE REGULARIZACIÓN

El entrenamiento de las GANs a menudo requiere de una regularización adicional, ya que son altamente inestables. Para estabilizar el entrenamiento, se han propuesto varias técnicas que incluyen el *instance noise* [27], la regularización Jensen-Shannon [47], las *gradient penalties* [50, 47], la normalización espectral [65], la regularización de defensa adversaria [10] y regularización de consistencia [85].

La regularización de consistencia (*consistency regularization*) es una técnica popular en la literatura de aprendizaje semi-supervisado que pretende mejorar el entrenamiento de un modelo. El principio de regularización de la consistencia subraya que los modelos de aprendizaje automático deberían ser menos sensibles a la perturbación extra impuesta a las muestras de entrada. Por tanto, el modelo debe dar resultados consistentes para una entrada y sus variantes perturbadas. Esto se consigue normalmente minimizando la diferencia entre la predicción sobre la entrada original y la predicción sobre la versión perturbada de la entrada.

Siguiendo esta línea de investigación, han surgido varios estudios como los de Sajjadi et al. (2016) [69], Laine y Aila (2016) [39], Zhai et al. (2019) [83], Xie et al. (2019) [81] y Berthelot et al. (2019) [8].

A continuación, se detallan distintos métodos que aplican técnicas de regularización de consistencia.

#### 3.3.2.1. CONSISTENCY REGULARIZATION FOR GANS (CR)

En 2019 Han Zhang et al. publican una técnica de aumento de la eficiencia de las GANs basado en regularización de la consistencia en su artículo titulado "*Consistency Regularization for Generative Adversarial Networks*" [85].

Se trata de un estabilizador de entrenamiento simple y efectivo basado en la noción de regularización de consistencia. En concreto, la técnica consiste en aumentar los datos de entrada del discriminador y penalizar la sensibilidad del discriminador a estos aumentos.

En los experimentos llevados a cabo en el artículo, se demuestra que la regularización de consistencia funciona eficazmente en varias arquitecturas GAN, funciones de pérdida y ajustes del optimizador. El método mejora las puntuaciones FID para la generación de imágenes en comparación con otros métodos de regularización en CIFAR-10 y CelebA.

### 3.3.2.2. A SIMPLE FRAMEWORK FOR CONTRASTIVE LEARNING OF VISUAL REPRESENTATIONS (SIMCLR)

En 2020 Ting Chen et al. publican una técnica de aumento de la eficiencia de las GANs basado en regularización de la consistencia en su artículo titulado "*A Simple Framework for Contrastive Learning of Visual Representations*" [16]. Los autores proponen un marco de aprendizaje contrastivo autosupervisado que pretende extraer eficazmente representaciones útiles sin necesidad de arquitecturas especializadas. El objetivo de esta técnica es maximizar la concordancia entre diferentes aumentos de la misma muestra mediante una función de pérdidas contrastiva en el espacio latente.

## 4. Métodos

En este apartado se detallarán los métodos estudiados en el trabajo. Las técnicas implementadas incluyen la arquitectura generativa adversaria BigGAN, la técnica de transferencia de conocimiento y el ajuste fino. Además, se aplican otros métodos con el propósito de aumentar la eficiencia de los modelos como el aumento de datos y las técnicas de regularización. El código desarrollado en el trabajo se encuentra disponible en <https://github.com/javier-marti-isasi/GANs-with-limited-data>.

### 4.1. METODOLOGÍA IMPLEMENTADA

El trabajo ha sido realizado mediante la implementación de la metodología CRISP-DM, desde la comprensión del negocio hasta la evaluación. Sus siglas corresponden a *Cross Industry Imaged Process for Data Mining* y es una metodología creada para dar forma a los proyectos de minería de datos (*data mining*). Fue concebida en 1996 y puesta en marcha por medio de un proyecto de financiación europea de 1997.

Integra todas las tareas necesarias desde la fase de comprensión del problema hasta la puesta en producción. Se trata de la metodología más utilizada y reconocida en la actualidad. Consta de 6 pasos para concebir un proyecto pudiendo tener iteraciones cíclicas según las necesidades de los desarrolladores. Los pasos presentes en la metodología son la comprensión del negocio, la comprensión de los datos, la preparación de los datos, el modelado, la evaluación y el despliegue.

## 4.2. ARQUITECTURA GAN

En los experimentos realizados en este trabajo se emplea la arquitectura BigGAN por sus resultados de gran calidad. Como se ha comentado en el punto 3.1.2., esta arquitectura fue publicada en 2018 por DeepMind (Andrew Brock, et al.) en el artículo "*Large Scale GAN Training for High Fidelity Natural Image Synthesis*" [11].

Siguiendo lo que los autores llaman "truco de truncamiento" (*truncation trick*), la distribución latente utilizada para el muestreo es diferente de la distribución  $z \sim N(0,1)$  utilizada durante el entrenamiento. Concretamente, la distribución utilizada durante el muestreo es una distribución normal truncada (remuestreo de  $z$  que tienen una magnitud superior a un determinado umbral). Cuanto menor sea el umbral de truncamiento, mayor será la verosimilitud de las muestras generadas, a costa de una menor variabilidad.

Como su nombre indica, BigGAN es una mejora respecto a SAGAN [84] en parte simplemente por ser más grande. BigGAN utiliza un tamaño de lote (batch size) de 2.048 (8 veces mayor que el tamaño de lote de 256 utilizado en SAGAN) y un tamaño de canal que se incrementa en un 50% en cada capa. Por otro lado, BigGAN muestra distintas técnicas de mejora como la inclusión de embeddings, la aplicación de regularización ortogonal y la incorporación del vector latente  $z$  en cada capa del generador, en lugar de sólo en la capa inicial.

## 4.3. TRANSFERENCIA DE CONOCIMIENTO Y AJUSTE FINO

La primera técnica implementada para mejorar la eficiencia de los modelos es el método de transferencia de conocimiento y ajuste fino.

Se plantea la aplicación del método definido en el modelo TransferGAN [76], pero se descarta por los problemas de colapso del modo y sobreajuste que sufre al actualizar todos los parámetros del generador para adaptarse al dominio de destino.

La técnica de transferencia de conocimiento propuesta en el artículo de Batch statistics adaptation [54] es menos susceptible al colapso del modo, pero reduce significativamente la flexibilidad de adaptación del modelo. Esto es debido a que al cambiar sólo los parámetros de la normalización por lotes permite los cambios de estilo, pero no se espera que funcione cuando hay que cambiar la forma. También sustituye la pérdida GAN por una pérdida de error cuadrático medio. Como resultado, su modelo sólo aprende la relación entre los vectores latentes y las muestras de entrenamiento dispersas, requiriendo que la distribución de ruido de entrada sea truncada durante la inferencia para generar muestras realistas.

Estos dos métodos requieren seleccionar manualmente la capa específica que se congela. Para evitar estos problemas, el diseño de los modelos en los que se aplica trasferencia por conocimiento en este TFM está inspirado en el artículo de MineGAN [78]. Este método pretende solucionar los problemas de TrasferGAN y Batch statistics adaptation al localizar automáticamente los pesos específicos objetivo.

En la figura 4.1 se representa la intuición detrás de MineGAN. La red minera desplaza la distribución de entrada hacia las regiones más prometedoras respecto a las imágenes reales.

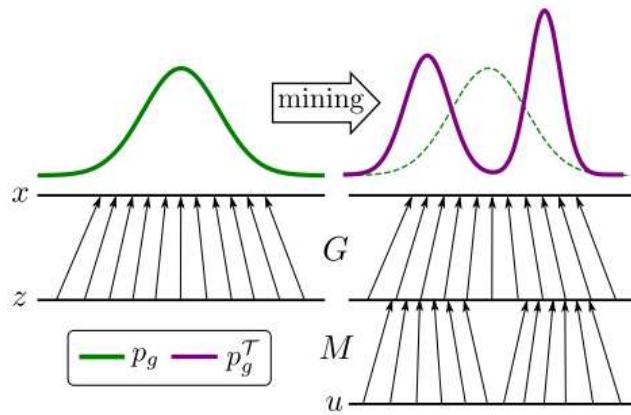


Figura 4.1: Intuición detrás de MineGAN [78].

Durante la primera etapa de entrenamiento (figura 4.2) de la red minera, el generador permanece fijo. En una segunda etapa, se aplica el ajuste fino en la red minera, en el generador y en el discriminador.

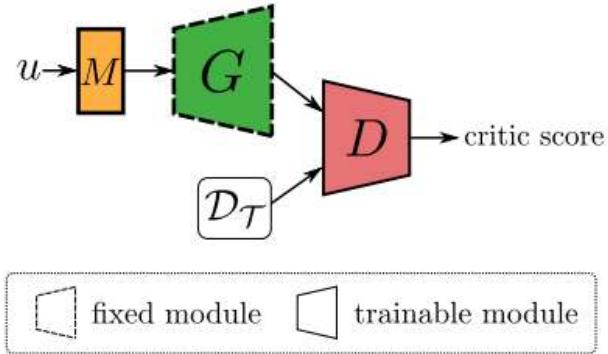


Figura 4.2: Esquema MineGAN en la primera etapa implementado en una sola red GAN [78].

El objetivo de la red minera es transformar la variable de ruido de entrada original  $u \sim p_z(u)$  de forma que se identifique mejor a las regiones de  $p_g(x)$  y que se ajuste mejor con la distribución objetivo.

El método completo consta de dos etapas. En la primera etapa sin ajuste fino (figura 4.2), se dirige el espacio latente del generador fijo  $G$  hacia zonas adecuadas para la distribución objetivo. En la segunda etapa se actualizan los pesos del generador a través del ajuste fino. Se ajustan las funciones de pérdida del discriminador y generador:

$$L_D^M = E_{u \sim p_z(u)} [D(G(M(u)))] - E_{x \sim p_{data}^T(x)} [D(x)],$$

$$L_G^M = -E_{u \sim p_z(u)} [D(G(M(u)))].$$

Los parámetros de  $G$  no se modifican durante el entrenamiento, pero los gradientes son retropropagados hasta  $M$ . Esta estrategia de entrenamiento orientará a la red minera hacia las regiones más prometedoras del espacio de entrada, es decir, las que generan imágenes cercanas a  $D_T$ .  $M$  extrae las regiones de entrada relevantes de  $p_z(u)$  y devuelve la distribución  $p_z^T(z)$  que se centrará en las regiones más adecuadas, ignorando otras que conducen a muestras alejadas de la distribución objetivo  $p_{data}^T(x)$ .

La red minera contiene relativamente pocos parámetros y es, por lo tanto, menos propensa al sobreajuste. Esto ocurre cuando se aplica ajuste fino directamente al generador  $G$  [54]. La transferencia de conocimiento al nuevo dominio se realiza aplicando ajuste fino tanto en la red minera  $M$  como en el generador  $G$ . Como consecuencia, el riesgo de sobreajuste disminuye, ya que la distribución generativa está más cerca del objetivo. Además, el entrenamiento es sustancialmente más eficiente que al realizar ajuste fino directamente a la red preentrenada [78], donde las imágenes sintéticas no son necesariamente similares a las muestras objetivo. El modelo preentrenado minado hace que el muestreo sea más eficaz, lo que conduce a gradientes menos ruidosos y una señal de entrenamiento más limpia.

## 4.4. OTRAS TÉCNICAS DE AUMENTO DE EFICIENCIA DE LAS GANS

Tras la implementación del método de transferencia de conocimiento y ajuste fino, se decide estudiar la técnica de aumento de datos y las técnicas de regularización. A continuación, se detallan cada una de ellas.

### 4.4.1. AUMENTO DE DATOS

Con la implementación de la técnica de aumento de datos se trata de mejorar aún más la eficiencia del modelo diseñado. Principalmente se pretende mejorar la estabilidad de los entrenamientos, minorar el sobreajuste y aumentar la calidad de las muestras generadas.

En este trabajo, el diseño de los modelos en los que se aplica aumento de datos está basado en el artículo DiffAugment [54]. Se plantea la implementación del modelo ADA [34], pero los autores no afirman en el artículo que su método es más eficiente que DiffAugment al no comprarlos en un marco común.

En intentos anteriores al artículo de DiffAugment, se aplican aumentos de datos únicamente a las muestras reales. Con este método se consiguen livianas mejorías al realizar ligeras transformaciones en las muestras reales. Pero realizar grandes transformaciones de los datos únicamente en las muestras reales, conlleva a problemas de distorsión o a una coloración extraña. Las salidas generadas estarán, por tanto, igualmente distorsionadas. Esto es debido a que el generador es forzado a coincidir con la distribución aumentada y distorsionada, no con la verdadera distribución.

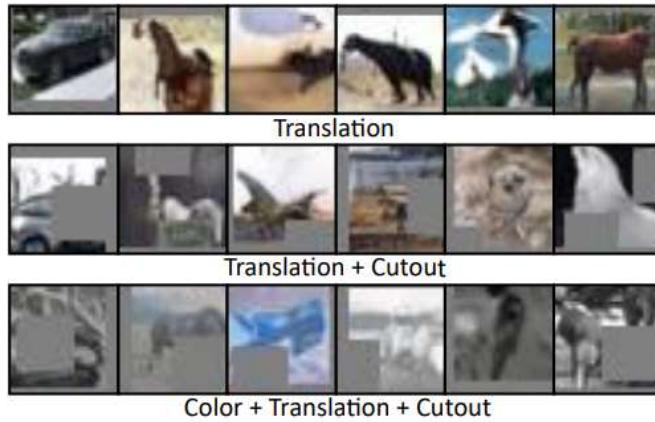


Figura 4.3: Imágenes generadas al aumentar sólo las muestras reales. Las mismas transformaciones aparecen en las imágenes generadas. [54].

En artículos posteriores se proponen técnicas de aumento de datos en las muestras reales y falsas. Sin embargo, en las primeras aproximaciones el discriminador no logra identificar las imágenes falsas que no están aumentadas, lo que lleva a una precisión inferior al 10%. Esto se debe a que el generador  $G$  recibe su gradiente de las imágenes falsas no aumentadas. Como resultado, se aprecian problemas en los modelos que aumentan las imágenes reales y sintéticas sin propagar correctamente los gradientes.

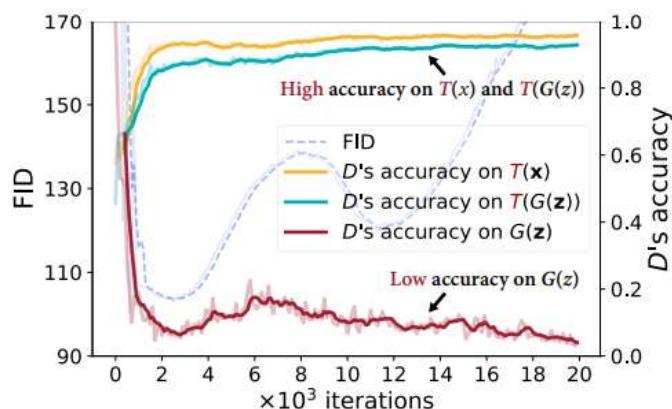


Figura 4.4: Precisión obtenida en estudios previos al aumentar las muestras reales y falsas [54]. La optimización desequilibrada entre  $G$  y  $D$  penaliza el entrenamiento.

Para combatir los problemas anteriores, el diseño de los modelos de este TFM está basado en el artículo de DiffAugment. Este método ofrece una solución que aumenta las imágenes reales y falsas utilizadas en la red discriminadora. Además, también propaga con éxito los gradientes de las muestras aumentadas a  $G$ .

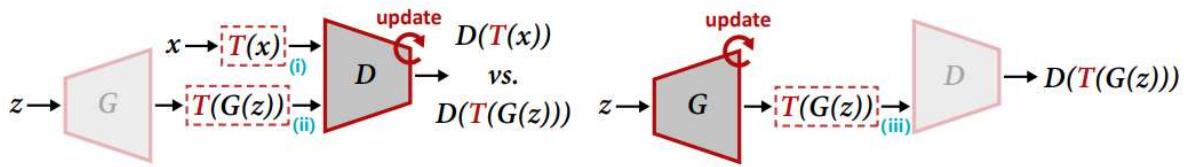


Figura 4.5: Resumen de la actualización de las redes en DiffAugment,  $D$  (izquierda) y  $G$  (derecha) [54]. Este método aplica el aumento  $T$  tanto a las muestras reales  $x$  como a las muestras de salida generadas  $G(z)$ . Cuando se actualiza  $G$ , los gradientes deben propagarse a través de  $T$ , lo que requiere que  $T$  sea diferenciable con respecto a la entrada.

Al propagar los gradientes, se consigue identificar las imágenes falsas que no han sido aumentadas. Para permitir la propagación del gradiente al generador  $G$ , simplemente se aseguran de que el aumento es, como su nombre indica, diferenciable. Las funciones de pérdidas del discriminador y del generador son las siguientes:

$$L_D = E_{(x \sim p_{data})(x)} [f_D(-D(x))] + E_{(z \sim p(z))} [f_D(D(G(z)))]$$

$$L_G = E_{z \sim p(z)} [f_G(-D(G(z)))]$$

#### 4.4.2. REGULARIZACIÓN

Tras el estudio de los métodos de transferencia de conocimiento, ajuste fino y aumento de datos, se pretende estudiar el método de regularización de consistencia para aumentar la estabilidad de los modelos diseñados. En este TFM, el diseño de los modelos en los que se aplican técnicas de regularización está basado en el artículo *Consistency Regularization* (CR) [85].

La clasificación del discriminador entre imágenes reales y falsas debe ser invariable para cualquier aumento válido de los datos específicos del dominio. Si se volteá una imagen horizontalmente o se traslada unos pocos píxeles, no debería cambiar el hecho de que una imagen sea real o no. Sin embargo, el discriminador no garantiza esta propiedad explícitamente. Para resolverlo, los autores del artículo CR proponen una regularización de consistencia en el discriminador durante el entrenamiento. Aumentan aleatoriamente las imágenes de entrenamiento que recibe el discriminador y penalizan la sensibilidad del discriminador a esos aumentos.

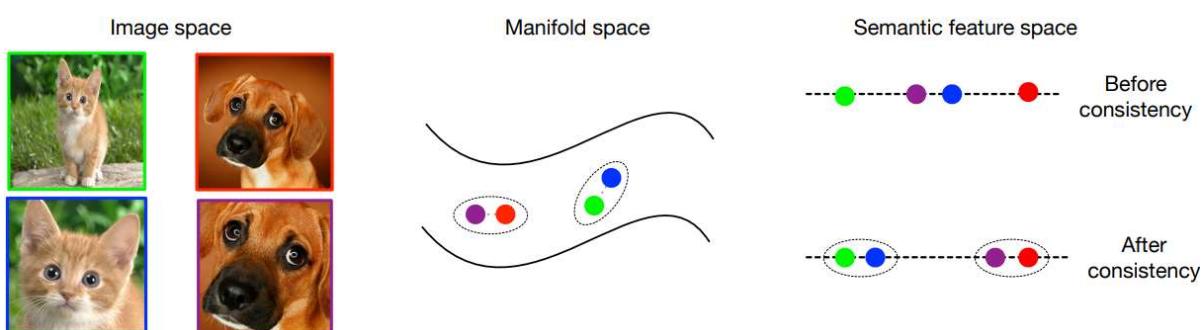


Figura 4.6: Representación de *Consistency Regularization* [85]. Antes de la regularización, el perro y el gato ampliados (abajo a la izquierda) pueden estar más cerca de sus imágenes originales en el espacio de características inducido por el discriminador. Esto se ilustra en la parte superior derecha (el espacio de características semánticas), donde el punto púrpura está más cerca del punto azul que del punto rojo. Despues de imponer la regularización de la consistencia, el punto púrpura se approxima más al punto rojo.

La propuesta de regularización viene dada por:

$$\min_D L_{cr} = \min_D \sum_{j=m}^n \lambda_j \| D_j(x) - D_j(T(x)) \|^2,$$

donde  $D_j(x)$  representa el vector de salida antes de la activación de la  $j^a$  capa del discriminador dada la entrada  $x$ .  $T(x)$  denota una función estocástica de aumento de datos.  $j$  indexa las capas,  $m$  es la capa inicial y  $n$  es la capa final en la que se aplica la regularización.  $\lambda_j$  es el coeficiente de peso de la  $j^a$  capa.

Esta regularización de la consistencia induce al discriminador a producir la misma salida a partir de varios aumentos de datos, preservando la semántica de la entrada.

# 5. Resultados

Este apartado contiene los resultados alcanzados en la implantación de los métodos anteriormente citados. En primer lugar, se detallan las limitaciones del *hardware* y las tecnologías empleadas. Seguidamente, se comentan las métricas de evaluación y conjuntos de datos utilizados. Finalmente, se detallan los modelos diseñados y los resultados obtenidos.

## 5.1. LIMITACIONES DE HARDWARE

Tal y como se ha comentado en el punto 1.3.3.1., el entrenamiento de las GANs es muy costoso desde el punto de vista computacional. Requiere GPUs de alta potencia y mucho tiempo de entrenamiento para conseguir resultados de calidad.

BigGAN requiere de 15 días para entrenar 150K iteraciones en 8 GPUs V100 usando el *dataset* ImageNet. Mediante el uso de técnicas como la transferencia de conocimiento, se permite mejorar y acelerar el entrenamiento de los modelos. Sin embargo, aún con esta implementación todavía es necesario un gran poder de computación para realizar entrenamientos en tiempos razonables.

Los entrenamientos de este trabajo han sido realizados mediante una GPU NVIDIA TITAN X de 12Gb. Hay que tener en cuenta que, para realizar los entrenamientos, la GPU utilizada tiene que ser compatible con las versiones de CUDA requeridas por cada modelo, CUDA 10 en el caso de los modelos empleados. Por desgracia, las GPUs GeForce RTX serie 30 de Nvidia no son compatibles con esta versión de CUDA. El procesador utilizado es un Intel® Core™ i7-5820K de 3.30GHz y la memoria RAM disponible es de 24Gb.

Google Colaboratory Pro fue utilizado en entrenamientos muy cortos, pero se descartó en el resto de los entrenamientos por exceder las limitaciones de uso.

En este TFM se diseñan modelos que pretenden obtener resultados de alta calidad reduciendo en gran medida los requerimientos computacionales. Sin embargo, el hardware disponible no alcanza los recursos mínimos recomendados por algunos modelos diseñados.

## 5.2. TECNOLOGÍAS EMPLEADAS

Los experimentos se han realizado en la distribución de Linux Ubuntu 18.04. Se han empleado principalmente dos entornos de desarrollo para satisfacer los modelos empleados. Los entornos corresponden a las versiones de CUDA 10.1 y 10.2. Se ha empleado el lenguaje de programación Python para todos los modelos, en las versiones Python 3.5 y Python 3.7.

Respecto a las librerías, se han utilizado principalmente PyTorch (versiones 1.4.0 y 1.0.1), TensorFlow (versiones 1.10.0 y 0.4.2), NumPy (versiones 1.14 y 1.20), OpenCV (versión 3.4.2), h5py (versiones 2.8 y 1.10), SciPy (versiones 1.1 y 1.2) y tqdm (versión 4.6.1).

## 5.3. MÉTRICAS DE EVALUACIÓN

En este apartado se hace un repaso de las métricas más utilizadas en la evaluación de las redes generativas adversarias y se comenta cuáles han sido utilizadas en este trabajo.

### 5.3.1. FUNCIONES DE PÉRDIDA

Las GANs están formadas por dos redes neuronales antagónicas que compiten entre sí en un constante juego de suma cero. Por lo que la ganancia o pérdida de una de las redes se compensa con la ganancia o pérdida de la opuesta. Como consecuencia, no se cumple que cuanto menor sea la pérdida del generador, mayor será la calidad de las muestras que produce. En lugar de ello, la pérdida del generador debe ser comparada con la del discriminador, que se encuentra en constante mejora. Por tanto, no es tan sencillo evaluar la mejora del modelo mediante estas métricas. Puede darse el caso de que el generador produzca muestras cada vez de mayor calidad, mientras que la función de pérdida va incrementando.

No obstante, como se indicó en el punto 5.3.1., las funciones de pérdida pueden ser utilizadas para comprobar si se producen problemas de no convergencia. Esto se aprecia cuando la función de pérdida del generador y discriminador oscilan sin lograr una estabilidad. En este trabajo se evaluarán estas métricas con el fin de analizar estos problemas en los entrenamientos.

### 5.3.2. INCEPTION SCORE (IS)

La métrica de *Inception Score* es muy utilizada en la evaluación de las GANs. Esta métrica trata de puntuar el realismo de los datos generados evaluando dos

cualidades deseables de las imágenes reales. Por un lado, la imagen debe contener objetos claros, no borrosos. La red de percepción debe tener una gran confianza en que la imagen generada pertenece a una clase determinada. Por otro lado, esta métrica evalúa la diversidad de las imágenes. En otras palabras, el generador debe producir imágenes que representen idealmente una etiqueta de clase diferente.

Este modo de evaluación penaliza en notablemente los experimentos realizados en este trabajo ya que se emplean conjuntos de datos de una única clase. Por tanto, esta métrica no se contempla para evaluar los resultados obtenidos.

### 5.3.3. FRÉCHET INCEPTION DISTANCE (FID)

La métrica de *Fréchet Inception Distance* (FID), también llamada distancia *Wasserstein-2*, es otra métrica muy utilizada en la evaluación de las GANs. Esta métrica, a diferencia del IS, tiene la ventaja de que comprara los datos reales con los datos generados.

EL FID comprara la media y la covarianza de las imágenes reales y las generadas sobre los mapas de características producidos al pasar las imágenes reales y generadas a través de una red Inception-v3 previamente entrenada. Esta evaluación es más favorable para un modelo cuanto más bajos sean los resultados de la métrica, ya que significa que las estadísticas de las imágenes generadas son muy similares a las de las imágenes reales. Se calcula como:

$$\text{FID} (X_1, X_2) = \left\| \mu_1 - \mu_2 \right\|_2^2 + \text{Tr} (\Sigma_1 + \Sigma_2 - 2(\Sigma_1 \Sigma_2)^{\frac{1}{2}}).$$

Esta es la métrica utilizada como referencia en todos los experimentos de este trabajo.

## 5.4. CONJUNTOS DE DATOS EMPLEADOS

Para la realización de los experimentos, se hace uso de dos conjuntos de datos. El primer *dataset* utilizado, **100-Obama**, se compone de 100 imágenes de 128x128px de la cara de Obama [88]. El segundo *dataset* utilizado, **1000-LSUN**, corresponde a una selección de 1000 imágenes del *dataset* LSUN de dormitorios [73, 16] a una resolución de 128x128px. En la preparación de los conjuntos de datos en formato hdf5, se aplica el método de aumento de datos implementado en BigGAN, que consiste en transformar las imágenes con volteos horizontales.

El número de muestras de entrenamiento es un factor muy determinante en el tiempo de entrenamiento. Aproximadamente, el tiempo de entrenamiento obtenido mediante los distintos modelos diseñados en el *dataset* 100-Obama oscilan entre 1-2 días. Mientras que el tiempo de entrenamiento del *dataset* 1000-LSUN oscila entre 9-16 días. En los primeros experimentos, se hace uso del *dataset* 100-Obama. Este conjunto de datos más reducido es utilizado en multitud de modelos que combinan distintas técnicas de entrenamiento. Una vez obtenidos los resultados de todos los modelos diseñados, se realiza un último entrenamiento empleando el *dataset* 1000-LSUN con el modelo más prometedor.

Incluso utilizando técnicas de transferencia de conocimiento, los modelos GAN requieren de muchas imágenes (al menos varios miles) para conseguir un entrenamiento estable y unos resultados de calidad. Empleando el *dataset* 100-Obama, con un número de imágenes extremadamente reducido, se pretende poner a prueba esta limitación de las GANs. Con los experimentos que emplean el *dataset* 1000-LSUN, se pretende testear los modelos con conjuntos de datos que presentan mucha mayor variabilidad.

El *dataset* 100-Obama se encuentra disponible en <https://data-efficient-gans.mit.edu/datasets/>. El *dataset* 1000-LSUN se encuentra disponible en [https://drive.google.com/drive/folders/1id\\_oaQUgqnt7o5fU9EDc6CFLK8DxwpGX](https://drive.google.com/drive/folders/1id_oaQUgqnt7o5fU9EDc6CFLK8DxwpGX)

## 5.5. MODELOS DISEÑADOS

En este apartado se explicarán con detalle los modelos diseñados y su parametrización. Se trata del modelo BigGAN desde cero y otros tres modelos diseñados en los que se aplican la técnica de transferencia de conocimiento y ajuste fino (TC), el método de aumento de datos (AD) y la técnica de regularización de consistencia (RC).

### 5.5.1. Modelo BigGAN (desde cero)

Este modelo se implementa con configuración por defecto de la red BigGAN con la excepción de la reducción del *batch size* de 256 a 24 por limitaciones de uso de memoria GPU, manteniendo el resto de las configuraciones. El número de acumulaciones del gradiente se mantiene en 8, tanto para el generador como para el discriminador y la tasa de aprendizaje (*learning rate*) se mantiene en  $1e^{-4}$  para el generador y  $4e^{-4}$  para el discriminador. El modelo se entrena durante 100 épocas.

### 5.5.2. Modelo BigGAN+TC

El método de transferencia de conocimiento se realiza por medio de la red Inception-V3, preentrenada en el *dataset* ImageNet a una resolución de 128x128px aplicando un *batch size* de 256 y un número de acumulaciones de gradiente de 8. Está entrenado en 138k iteraciones, justo antes de colapsar, obteniendo un TF *Inception Score* de 97.35. En los experimentos realizados, se han obtenido mejores resultados en esta red que en la red Inception-V3 entrenada hasta 100K iteraciones, a pesar de estar más alejada del colapso y, por tanto, poder ser más adecuada para realizar ajuste fino. El modelo preentrenado utilizado está disponible en <https://drive.google.com/file/d/1nAle7FCVFZdix2--ks0r5JBkFnKw8ctW/view>.

Se mantienen las configuraciones de BigGAN por defecto a excepción del *batch size*, que se reduce de 256 a 24. También se modifica el número de acumulaciones del gradiente tanto para el generador como para el discriminador, que se aumenta de 8 a 16, mejorando levemente los resultados obtenidos. La tasa de aprendizaje se mantiene en  $1e^{-4}$  para el generador y  $4e^{-4}$  para el discriminador. El modelo se entrena durante 100 épocas, pero el número de iteraciones por época disminuye al aumentar el número de acumulaciones del gradiente.

### 5.5.3. Modelo BigGAN+TC+AD

Una vez aplicado y ajustado el método de transferencia de conocimiento a la red BigGAN, se aplica el método de aumento de datos sobre el mismo. Se mantiene el *batch size* de 24 del modelo BigGAN+TC, pero se modifica el número de acumulaciones del gradiente, que se restablece en 8 como en la red BigGAN por defecto. También se modifica la tasa de aprendizaje del generador, que se modifica de  $1e^{-4}$  a  $2e^{-4}$ , y la del discriminador, que se modifica de  $4e^{-4}$  a  $2e^{-4}$ . Se entrena el modelo con el *dataset* 100-Obama durante 100 épocas en el experimento de 8K iteraciones y durante 150 épocas en el experimento de 18K iteraciones. El entrenamiento realizado con en el *dataset* 1000-LSUN fue programado para una duración de 150 épocas, pero se detuvo en la iteración 34K al no observar mejoría en el entrenamiento.

Se mantiene el método de aumento de datos de las imágenes reales implementado por defecto en la arquitectura BigGAN. Tanto en los entrenamientos realizados con el modelo BigGAN (desde cero) como con el modelo BigGAN+TC, se realiza aumento de datos en las muestras reales a través de volteos horizontales de las imágenes. En el modelo BigGAN+TC+AD se aplica el método de aumento de datos en las muestras reales y falsas, como se explica en este documento en el punto 4.4.1.

Se realizan aumento de datos a partir de transformaciones de recorte, traslaciones y ajustes de color (brillo, saturación y contraste). Estos aumentos corresponden a la configuración recomendada en el artículo de DiffAugment para *datasets* de pocas imágenes (cientos de muestras). Se comprueba que, efectivamente, se obtienen mejores resultados que aplicando únicamente volteos horizontales.

En la implementación de este método, se tuvo en cuenta la adecuada actualización de los pesos en la red minera encargada de la transferencia de conocimiento, además de la red generadora y discriminadora tal y como se muestra en la figura 5.1.

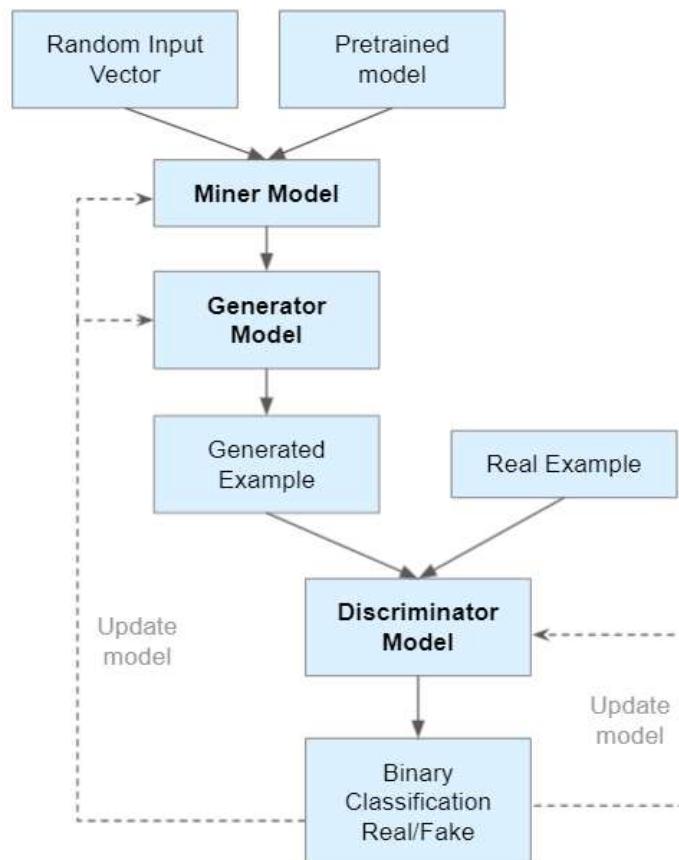


Figura 5.1: Representación esquemática del modelo BigGAN+TC+AD. Fuente propia.

#### 5.5.4. Modelo BigGAN+TC+AD+RC

Una vez aplicado y ajustado el método de aumento de datos sobre el diseño de BigGAN+TC+AD, se decide implementar la técnica de regularización de consistencia. Por limitaciones en el uso de la RAM, se tuvo que reducir el *batch size* de 24 a 16, lo que dificultó la comparación posterior con el resto de los modelos diseñados.

Partiendo del modelo BigGAN+TC+AD, se mantuvo la configuración de la tasa de aprendizaje de  $2e^{-4}$ , tanto para el generador como para el discriminador, y el número de acumulaciones del gradiente de 8. Sin embargo, se modifica la configuración del aumento de datos eliminando las transformaciones de color. Este ajuste consigue una cierta mejoría en los resultados.

El modelo se entrena durante 100 épocas para el experimento entrenado en un máximo de 8K iteraciones y durante 150 épocas para el experimento entrenado en un máximo 18K iteraciones.

## 5.6. EXPERIMENTOS Y RESULTADOS

En este apartado se exponen una selección de los experimentos realizados y sus resultados.

En primer lugar, se prueba la técnica de transferencia de conocimiento sobre el *dataset* 100-Obama. Para ello, se realiza un entrenamiento con la red BigGAN desde cero y se compraran los resultados con el modelo que incluye el método de transferencia de conocimiento y ajuste fino (TC). Después, se trata de mejorar los resultados diseñando nuevos modelos que incluyen las técnicas de aumento de datos (AD) y de regularización de consistencia (RC). Por último, se entrena el modelo más prometedor con el *dataset* 1000-LSUN, de mayor variabilidad de muestras, y se compraran los resultados con el *dataset* 100-Obama.

En la siguiente tabla se exponen los resultados obtenidos con los distintos modelos diseñados, entrenados en el *dataset* 100-Obama. Para su adecuada comparación, los entrenamientos tienen un máximo de 8K iteraciones:

### Dataset 100-Obama - máx. 8K iteraciones

Modelo GAN	Método			FID
	(TC)	(AD)	(RC)	
BigGAN (desde cero)				343.95
BigGAN+TC	✓			50.96
BigGAN+TC+AD	✓	✓		37.90
BigGAN+TC+AD+RC	✓	✓	✓	56.86

Tabla 5.1: Resultados obtenidos de distintos entrenamientos con el *dataset* 100-Obama. Se prueba la arquitectura BigGAN desde cero y con la aplicación de los métodos de transferencia de conocimiento (TR), aumento de datos (AD) y regularización de consistencia (RC). Se obtiene el mejor FID alcanzado por cada entrenamiento en un máximo de 8K iteraciones.

Para los siguientes experimentos, se seleccionan los modelos BigGAN+TC+AD y BigGAN+TC+AD+RC y se aumenta la duración de los entrenamientos tomando el mismo *dataset* 100-Obama. Para su adecuada comparación, los entrenamientos tienen un máximo de 18K iteraciones.

### **Dataset 100-Obama - máx. 18K iteraciones**

Modelo GAN	Método			FID
	(TC)	(AD)	(RC)	
BigGAN+TC+AD	✓	✓		36.95
BigGAN +TC+AD+RC	✓	✓	✓	52.82

Tabla 5.2: Resultados obtenidos en distintos entrenamientos con el *dataset* 100-Obama. Se prueba la arquitectura BigGAN con la aplicación de los métodos de transferencia de conocimiento (TR), aumento de datos (AD) y regularización de consistencia (RC). Se obtiene el mejor FID alcanzado por cada entrenamiento en un máximo de 18K iteraciones.

Nota: Debido a un mayor consumo de RAM en el modelo BigGAN+TC+AD+RC, el *batch size* se tuvo que reducir de 24 a 16. Para favorecer la comparación de resultados, se entrenó durante 18K iteraciones el modelo BigGAN+TC+AD+RC y durante 12K iteraciones el modelo BigGAN+TC+AD.

Por último, se selecciona el modelo con el mejor rendimiento, BigGAN+TC+AD, y se testea en el *dataset* 1000-LSUN. Se comparan los resultados del modelo BigGAN+TC+AD mediante el entrenamiento con los *datasets* 100-Obama y 1000-LSUN durante 150 épocas.

## **Dataset 100-Obama + 1000-LSUN - 150 épocas**

Modelo GAN	<i>Dataset</i>	Método			FID
		(TC)	(AD)	(RC)	
BigGAN+TC+AD	100-Obama	✓	✓		36.95
BigGAN+TC+AD	1000-LSUN	✓	✓		78.43

Tabla 5.3: Resultados obtenidos con el *dataset* 100-Obama y 1000-LSUN. Se prueba la arquitectura BigGAN con la aplicación de los métodos de transferencia de conocimiento (TR) y aumento de datos (AD). Se obtiene el mejor FID alcanzado por cada entrenamiento durante 150 épocas.

Nota: Ambos entrenamientos tienen un máximo de 150 épocas. Sin embargo, el número de iteraciones por época necesario en el entrenamiento con el *dataset* 1000-LSUN es diez veces superior al del entrenamiento con el *dataset* 100-Obama por el mayor número de muestras del conjunto de datos.

A continuación, se incluyen muestras generadas por cada modelo entrenado con el *dataset* 100-Obama. Las imágenes corresponden a la iteración con el mejor FID alcanzado.



100-Obama BigGAN desde cero máx 8K it. FID:343.95	100-Obama BigGAN +TC máx 8K it. FID:50.96	100-Obama BigGAN +TC+AD máx 8K it. FID: <b>37.90</b>	100-Obama BigGAN +TC+AD+CR máx 8K it. FID:56.86	100-Obama BigGAN +TC+AD máx 18K it. FID: <b>36.95</b>	100-Obama BigGAN +TC+AD+CR máx 18K iter. FID:52.82
---	---	--	---	---	--

Figura 5.2: Imágenes generadas con distintos modelos usando el *dataset* 100-Obama.

A continuación, se incluyen muestras generadas por el modelo BigGAN+TC+AD en el *dataset* 1000-LSUN. Las imágenes corresponden a la iteración con el mejor FID alcanzado.

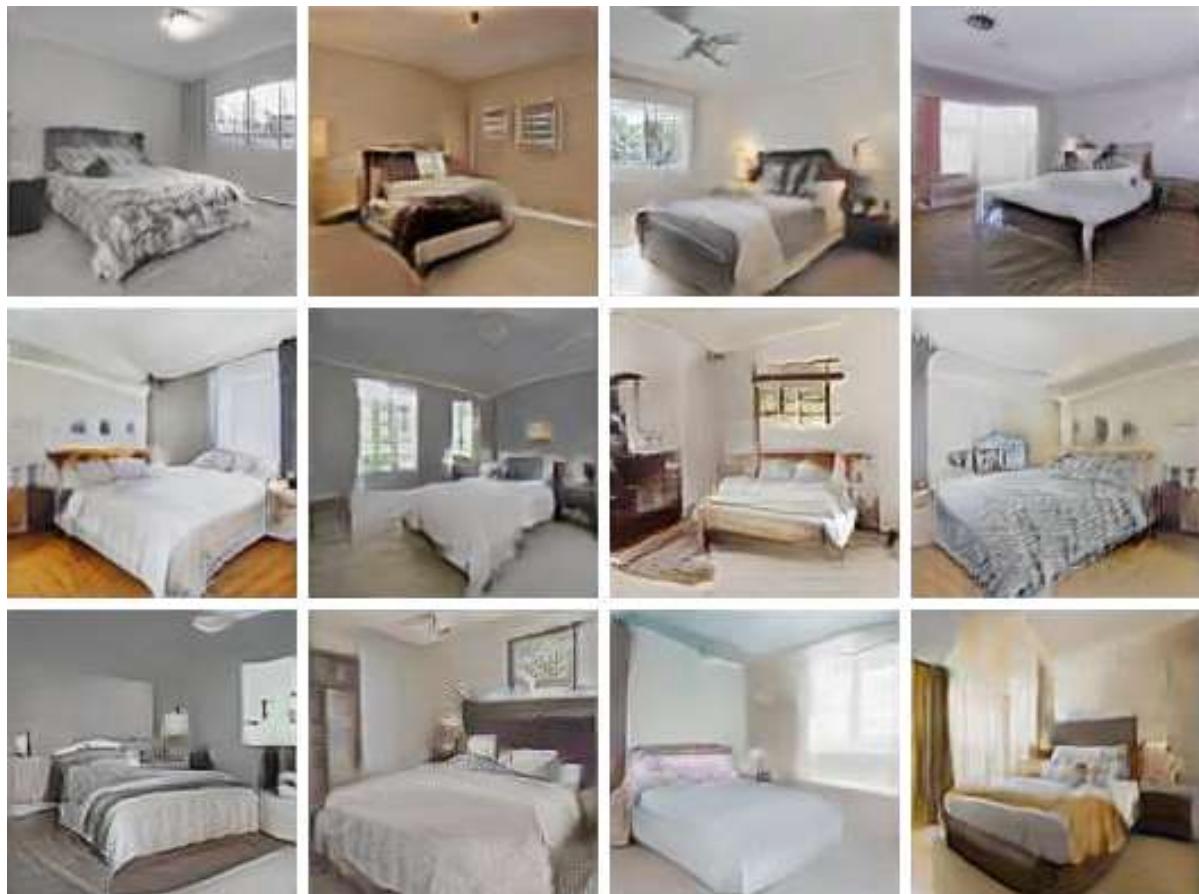


Figura 5.3: Imágenes generadas en el entrenamiento del modelo BigGAN+TC+AD usando el *dataset* 1000-LSUN durante 150 épocas. FID:78.43.

A continuación, se muestran imágenes generadas en diferentes fases del entrenamiento del modelo BigGAN+TC+AD empleando el *dataset* 100-Obama.



Figura 5.4: Imágenes generadas con el modelo BigGAN+TC+AD usando el *dataset* 100-Obama en diferentes iteraciones (de 0 a 8000, en saltos de 1000).

Un vídeo que muestra la evolución del entrenamiento del modelo BigGAN+TC+AD empleando el conjunto de datos 100-Obama está disponible en el siguiente enlace: <https://www.youtube.com/watch?v=Unf-7sDUdZ0>.

## 6. Conclusiones

En este trabajo se han alcanzado los objetivos propuestos. Se ha conseguido aumentar la eficiencia de las redes generativas adversarias mediante el método de transferencia de conocimiento y ajuste fino. Además, gracias a la aplicación de nuevas técnicas y al ajuste de la parametrización, se han mejorado los resultados. Los nuevos modelos diseñados han demostrado que la combinación de la técnica de transferencia de conocimiento y ajuste fino con métodos de aumento de datos y regularización de consistencia mejoran la estabilidad de los entrenamientos, las métricas y la calidad de las muestras generadas. Todo ello se ha conseguido en conjuntos de datos increíblemente pequeños, lo que amplía los horizontes de aplicabilidad de las redes generativas.

En los experimentos realizados se ha demostrado la enorme utilidad de la técnica de transferencia de conocimiento y ajuste fino, especialmente en *datasets* con pocas muestras y de un dominio similar al de la red preentrenada. Los resultados obtenidos con el entrenamiento de BigGAN desde cero han demostrado la ineficiencia del modelo en conjuntos de datos pequeños. Ni siquiera ha sido capaz de generar ciertas formas geométricas similares a las muestras reales. Esta ineficiencia queda reflejada en la gráfica A1.1 del Anexo 1, donde el generador parece incapaz de engañar al discriminador. Se ha comprobado que la red preentrenada en ImageNet durante 138K iteraciones tiene una mayor eficiencia que la preentrenada durante 100K iteraciones, a pesar de encontrarse próxima al colapso.

Otra técnica que ha demostrado el aumento de la eficiencia de las GANs, especialmente en conjuntos de datos pequeños, es el método de aumento de datos. Se ha comprobado que en entrenamientos con pocas muestras es más eficiente realizar fuertes transformaciones que incluyen traslaciones, recortes y

ajustes de color. Se ha comprobado que se mejora la eficiencia de los modelos al incrementar las imágenes reales y falsas y actualizar adecuadamente las redes neuronales. Además, se ha comprobado el aumento de la eficiencia de los entrenamientos al combinar los métodos de transferencia de conocimiento, ajuste fino y aumento de datos en un único modelo.

La técnica de regularización de consistencia en el modelo BigGAN+TC+AD+RC no ha conseguido mejorar el FID, en parte debido, muy probablemente, por la imposibilidad de realizar una parametrización igual al modelo BigGAN+TC+AD, por no satisfacer el aumento de RAM necesario. De hecho, el artículo sobre la técnica de regularización de consistencia SimCLR [16] advierte de la necesidad de aumentar el *batch size* y el tiempo de entrenamiento para conseguir la eficiencia del método. A pesar de ello, comparando las funciones de pérdidas en gráficas A1.5 y A1.6 del Anexo 1, se observan ciertas mejoras en la implementación. Se aprecia una mayor estabilidad del modelo BigGAN+TC+AD+RC y, comparando la evolución del FID en las gráficas A2.5 y A2.6 del Anexo 2, se observa claramente que, mientras que en el del modelo BigGAN+TC+AD el FID baja rápidamente y se mantiene constante, en el modelo BigGAN+TC+AD+RC el FID desciende poco a poco de forma más suave. Esto podría indicar que, alargando el entrenamiento, se podría seguir disminuyendo el FID hasta el punto de mejorar el del modelo BigGAN+TC+AD.

La eficiencia de las redes generativas está muy condicionada por la cantidad de muestras del *dataset*. Sin embargo, los modelos han funcionado muy satisfactoriamente en los conjuntos de datos utilizados con la implementación de las técnicas estudiadas.

En la comparación de los resultados obtenidos con los distintos *datasets*, se aprecia que una alta variabilidad de las muestras penaliza notablemente el rendimiento de los modelos, de ahí la gran importancia de partir de un conjunto de datos de alta calidad. Siendo entrenado en el mismo modelo con una duración

igual de 150 épocas, los resultados obtenidos con el *dataset* 1000-LSUN han sido peores que con el *dataset* 100-Obama, a pesar de que el conjunto de datos 1000-LSUN incluye 10 veces más de muestras reales y requiere un tiempo de entrenamiento casi 10 veces superior.

Por otro lado, se puede considerar que el aumento de la eficiencia conseguido con la técnica de transferencia de conocimiento y ajuste fino ha tenido un menor efecto en el *dataset* 1000-LSUN. Esta técnica, como se observa en los resultados, es mucho más eficiente cuanto más similares sean el dominio del *dataset* con el de las redes preentrenadas. En los experimentos se ha empleado la red Inception-V3 que ha sido entrenada en el *dataset* ImageNet. En este conjunto de datos se tiene especial relevancia las clases de persona. De las aproximadamente 20000 subclases que se incluyen en el *dataset*, existen 2832 categorías de personas. Gracias a ello, la red utilizada es especialmente eficaz en modelos entrenados con imágenes de caras de personas.

Como aspecto negativo, se aprecia una similitud de las muestras generadas con las muestras reales. Esta circunstancia es una situación común en las GANs que podría ser mejorada en futuras implementaciones, como se expone en el punto de líneas futuras.

En definitiva, se ha demostrado que las técnicas de transferencia de conocimiento, de ajuste fino, de aumento de datos y de regularización, amplían en gran medida la eficiencia de las GANs. Extiende su aplicabilidad a un mayor número de situaciones y lo aproxima al público general al reducirse enormemente la cantidad de datos y de poder computacional necesarios.

## 7. Líneas futuras

El mundo de las redes generativas adversarias en particular, y el de las redes generativas en general, está en constante evolución. Desde que surgieron, han sido ampliamente estudiadas debido al enorme potencial de sus aplicaciones. Cada semana se publica una considerable cantidad de artículos científicos relacionados con este campo. Se han conseguido grandes avances en los últimos años, desde nuevos métodos que consiguen aumentar la eficiencia de redes existentes, hasta nuevas arquitecturas revolucionarias que aspiran a convertirse en el nuevo paradigma de referencia.

Tal y como se ha demostrado en este trabajo, mediante el uso de técnicas de transferencia de conocimiento, de ajuste fino, de aumento de datos y de métodos de regularización, se ha conseguido aumentar en gran medida la eficiencia de las GANs. No obstante, se ha comprobado que, en conjuntos de datos con mayor variabilidad, la eficiencia de las redes se reduce. Además, aunque se consigue mejoría, en los modelos diseñados siguen apareciendo problemas como la inestabilidad, el colapso del modo o la verosimilitud de las muestras generadas con las muestras reales.

Siguiendo esta línea de investigación, en futuras implementaciones se podría continuar con el estudio de distintas técnicas que ayuden a minorar estos problemas. Tal y como exponen los autores en el artículo BigGAN, cuanto menor sea el umbral de truncamiento, mayor será la verosimilitud de las muestras generadas a costa de una menor variabilidad. Se podría realizar una investigación sobre el umbral de truncamiento, tratando de conseguir una mayor variabilidad sin reducir la calidad de las muestras generadas.

Una de las técnicas de transferencia de conocimiento y ajuste fino comentadas en el apartado del estado del arte y que podría ser estudiada, es el método

Freeze-D [52]. Sus autores afirman que este modelo supera a los anteriores a pesar de su simplicidad. En resultados observados en distintos artículos [52, 34, 78] se observa una mejoría de los resultados obtenidos respecto al método MineGAN [78] en función del conjunto de datos y de la red preentrenada. Esto es presumiblemente debido a una posible menor eficiencia del método de MineGAN cuando la distribución de origen y destino difieren.

Con el fin de mejorar la estabilidad de las redes GAN, se podría profundizar en la aplicación de otras técnicas de regularización de consistencia estudiando otros métodos como SimCLR [29]. Sin embargo, existen muchas otras técnicas de regularización que se podrían analizar como *instance noise* [27], regularización Jensen-Shannon [47], *gradient penalties* [50, 47], normalización espectral [65] y regularización de defensa adversaria [10]. Respecto al método de aumento de datos, se podría tratar de mejorar su rendimiento mediante el uso de unas transformaciones más diversas o con regularizaciones contrastadas [89]. También sería interesante estudiar el método ADA [34].

Por otro lado, se podrían estudiar distintas arquitecturas GAN, como StyleGAN2 [36], analizando cómo influyen los distintos métodos expuestos en este trabajo sobre esta red.

Otra línea futura de investigación podría centrarse en la aplicación de *transformers* [57]. Se trata de una técnica que pretende mejorar los resultados obtenidos por otras técnicas combinando arquitecturas del campo de la imagen y del campo del procesamiento del lenguaje natural. Siguiendo esta línea de investigación, se podría continuar analizando el método TransGAN [31]. Se trata de un método basado en GAN, pero con un generador y un discriminador creado a partir de *transformers*, sin la aplicación de convoluciones. TransGAN necesita una serie de técnicas de entrenamiento adicionales para conseguir resultados competitivos. Estas técnicas son el aumento de datos, la aplicación de entrenamientos multitarea y la inicialización localizada. Los autores aseguran que

es un método comparable con las GANs convolucionales y plantean cinco puntos de acción específicos para continuar con su perfeccionamiento.

Por último, se considera la posibilidad de publicar un artículo que condense la investigación realizada en este TFM.

## 8. Bibliografía

- [1] Amini, A y Soleimany, A. (2020). *Introduction to Deep Learning: deep generative models*. MIT EECS.
- [2] Anónimo. (s.f.) *Advances in Generative Adversarial Networks (GANs)*. BeyondMinds. Consultado el 14 de octubre de 2021. URL <https://beyondminds.ai/blog/advances-in-generative-adversarial-networks-gans>.
- [3] Anónimo. (2019). *GANs for Image Generation: ProGAN, SAGAN, BigGAN, StyleGAN*. CV Notes. URL <https://cvnote.ddlee.cc/2019/09/15/progan-sagan-biggan-stylegan>.
- [4] Antoniou, A., Storkey, A., y Edwards, H. (2017). *Data Augmentation Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1711.04340>.
- [5] Arjovsky, M. y Bottou, L. (2017). *Towards Principled Methods for Training Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1701.04862>.
- [6] Arjovski, M., Chintala, S., y Bottou, L. (2017). *Wasserstein GAN*. arXiv. URL <https://arxiv.org/abs/1701.07875>.
- [7] Asensio Cortés, G. (2018). Aprendizaje supervisado. Universidad Internacional de Valencia.
- [8] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., y Raffel, C. (2019). *MixMatch: A Holistic Approach to Semi-Supervised Learning*. arXiv. URL <https://arxiv.org/abs/1905.02249>.

- [9] Bonn, D. (2021). *An interview with David Bonn, computer vision and wildfire detection expert by Adrian Rosebrock*, PyImageSearch. URL <https://www.pyimagesearch.com/2021/10/13/an-interview-with-david-bonn-computer-vision-and-wildfire-detection-expert>.
- [10] Brady, Z y Krahnenb, P. (2019). *Don't let your discriminator be fooled*. En *International Conference on Learning Representations (ICLR)*.
- [11] Brock, A., Donahue, J., y Simonyan, K. (2018). *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. arXiv. URL <https://arxiv.org/abs/1809.11096>.
- [12] Brownlee, J. (2017). *How to Develop an Encoder-Decoder Model for Sequence-to-Sequence Prediction in Keras*. Towards data science. URL <https://machinelearningmastery.com/develop-encoder-decoder-model-sequence-sequence-prediction-keras>.
- [13] Brownlee, J. (2019). *18 Impressive Applications of Generative Adversarial Networks (GANs)*. Machine Learning Mastery. URL <https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks>.
- [14] Brownlee, J. (2019). *Tour of Generative Adversarial Network Models*. Towards data science. URL <https://machinelearningmastery.com/tour-of-generative-adversarial-network-models>.
- [15] Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G. C., Steinhardt, J., Flynn, C., Eigeartaigh, S., Beard, S., Belfield, H., Farquhar, S., Lyle, C., Crootof, R., Evans, O., Page, M., Bryson, J., Yampolskiy, R., y Amodei, D. (2018). *The Malicious Use of Artificial*

*Intelligence: Forecasting, Prevention, and Mitigation* arXiv. URL <https://arxiv.org/ftp/arxiv/papers/1802/1802.07228.pdf>.

- [16] Chen, T., Kornblith, S., Norouzi, M., y Hinton, G. (2020). *A Simple Framework for Contrastive Learning of Visual Representations*. arXiv. URL <https://arxiv.org/abs/2002.05709>.
- [17] Colomer Granero, A. y Muñoz Ríos, G. E. (2020). Redes neuronales y aprendizaje profundo. Universidad Internacional de Valencia.
- [18] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., y Darrell, T. (2014). *A deep convolutional activation feature for generic visual recognition*, en ICML, pp. 647–655.
- [19] Frantzell, F. (2019). *Deepfakes and the world of Generative Adversarial Networks*. Medium. URL <https://medium.com/@lennartfr/deepfakes-and-the-world-of-generative-adversarial-networks-bf6937e70637>
- [20] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., y Greenspan, H. (2018). *GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification*. arXiv. URL <https://arxiv.org/abs/1803.01229>.
- [21] Forsyth, D., y Ponce, J. (2012). *Computer Vision: A Modern Approach*. Pearson Education, Inc.
- [22] Foster, D. (2019). *Generative Deep Learning. Chapter 1: Generative Modeling*. O'Reilly Media, Inc. URL <https://www.oreilly.com/library/view/generative-deep-learning/9781492041931/ch01.html>.
- [23] Fuentes Hurtado, F. J. Aprendizaje no supervisado. (2019). Universidad Internacional de Valencia.

- [24] Goodfellow, I. (2016). *NIPS 2016 tutorial: Generative adversarial networks*. arXiv. URL <https://arxiv.org/abs/1701.00160>.
- [25] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., y Bengio, Y. (2014). *Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1406.2661>.
- [26] Gujar, S. (2018). *GANs in Tensorflow (Part II)*. Medium. URL <https://medium.com/@sanketgujar95/gans-in-tensorflow-261649d4f18d>
- [27] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., y Caurville, A. C. (2017). *Improved training of wasserstein gans*. En *Conference on Neural Information Processing Systems (NeurIPS)*.
- [28] Hoffman, J., Tzeng, E., Darrell, T., y Saenko, K. (2015). *Simultaneous deep transfer across domains and tasks*. CVPR, pp. 4068– 4076.
- [29] Howard, J. (2016). *LSUN bedroom scene 20% sample*. Kaggle. URL [https://www.kaggle.com/jhoward/lsvn\\_bedroom](https://www.kaggle.com/jhoward/lsvn_bedroom).
- [30] Janetzky, P. (2019). *Generative Networks: From AE to VAE to GAN to CycleGAN. Towards data science*. URL <https://towardsdatascience.com/generative-networks-from-ae-to-vae-to-gan-to-cyclegan-b21ba99ab8d6>.
- [31] Jiang, Y., Chang, S., y Wnag, Z. (2021). *TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up*. arXiv. URL <https://arxiv.org/abs/2102.07074>.
- [32] Karpathy, A., Abbeel, P., Brockman, G., Chen, P., Cheung, V., Duan, R., Goodfellow, I., Kingma, D., Ho, J., Hourttoft, R., Salimans, T., Schulman, J., Sutskever, I., y Zaremba, W. (2016). *Generative Models*. OpenAI Inc. URL <https://openai.com/blog/generative-models>.

- [33] Karras, T., Aittala, M., Laine, S., Häkkinen, R., Hellsten, J., Lehtinen, J., y Aila, T. (2021). *Alias-Free Generative Adversarial Networks (StyleGAN3)*, NVIDIA Labs. URL <https://nvlabs.github.io/stylegan3>.
- [34] Karras, T., Aittala, M., Laine, S., Hellsten, J., Lehtinen, J., y Aila, T. (2020). *Training generative adversarial networks with limited data*. arXiv. <https://arxiv.org/abs/2006.06676>.
- [35] Karras, T., Laine, S., y Aila, T. (2018). *A Style-Based Generator Architecture for Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1812.04948>.
- [36] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., y Aila, T. (2019). *Analyzing and Improving the Image Quality of StyleGAN*. arXiv. URL <https://arxiv.org/abs/1912.04958>.
- [37] Karras, T., Laine, S., Hellsten, J., Lehtinen, J., y Aila, T. (2017). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. arXiv. URL <https://arxiv.org/abs/1710.10196>.
- [38] Kingma, D.P. y Welling, M. (2013). *Auto-Encoding Variational Bayes*. arXiv. URL <https://arxiv.org/abs/1312.6114>.
- [39] Laine, S. y Aila, T. (2016). *Temporal Ensembling for Semi-Supervised Learning*. arXiv. URL <https://arxiv.org/abs/1610.02242>.
- [40] LeCun, Y. (s.f.). *LeNet-5, convolutional neural networks*. Consultado el 7 de octubre de 2021. URL <http://yann.lecun.com/exdb/lenet>.
- [41] Ledig, G., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., y Shi, W. (2016). *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. arXiv. URL <https://arxiv.org/abs/1609.04802>.

- [42] Li, F., Johnson, J., y Yeung, S. (2017). *Lecture 11: Detection and Segmentation*. Stanford University. URL [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture11.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf).
- [43] Liu, M y Tuzel, O. (2017). *Coupled Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1606.07536>.
- [44] Mahapatra, D., y Bozorgtabar, B. (2019). *Progressive Generative Adversarial Networks for Medical Image Super resolution*. arXiv. URL <https://arxiv.org/abs/1902.02144>.
- [45] Mantri, N. (s.f.) *Applications of Autoencoders*. OpenGenus Foundation. Consultado el 9 de octubre de 2021. URL <https://iq.opengenus.org/applications-of-autoencoders>.
- [46] Melchior, L. (2019). *Data Augmentation with GANs for Defect Detection*. Dida. URL <https://dida.do/blog/data-augmentation-with-gans-for-defect-detection>.
- [47] Mescheder, L, Geiger, A., y Nowozin, S. (2018). *Which training methods for gans do actually converge?* En *International Conference on Machine Learning (ICML)*.
- [48] Metz, L., Poole, B., Pfau, D., y Sohl-Dickstein, J. (2017). *Unrolled Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1611.02163>.
- [49] Mihajlovic, I. (2019). *Everything you ever wanted to know about Computer Vision. Towards data science*. URL <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>.

- [50] Miyato, T., Kataoka, T., Koyama, M., y Yoshida, Y. (2018). *Spectral Normalization for Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1802.05957>.
- [51] Mirza, M. y Osindero, S. (2014). *Conditional Generative Adversarial Nets*. arXiv. URL <https://arxiv.org/abs/1411.1784>.
- [52] Mo, S., Cho, M., y Shin, J. (2020). *Freeze the discriminator: a simple baseline for fine-tuning GANs*. arXiv. URL <https://arxiv.org/abs/2002.10964>.
- [53] Nair, N. (2020). *Data-Efficient GANs!*. Towards data science. URL <https://towardsdatascience.com/data-efficient-gans-d10acd595361>.
- [54] Noguchi, A. y Harada, T. (2019). *Image generation from small datasets via batch statistics adaptation*. arXiv. <https://arxiv.org/abs/2006.10738>
- [55] Oquab, M., Bottou, L., Laptev, I., y Sivic, J. (2014). *Learning and transferring mid-level image representations using convolutional neural networks*, en CVPR. IEEE. pp. 1717–1724.
- [56] Pan, S. J. y Yang, Q. (2010). *A survey on transfer learning*. TKDE, vol. 22, no. 10, pp. 1345–1359.
- [57] Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., y Tran, D. (2018). *Image Transformer*. arXiv. URL <https://arxiv.org/abs/1802.05751>.
- [58] Park, S., Jae-Sub, K., Jun-Ho, H., y Jong-Chan, K. (2021). *Review on Generative Adversarial Networks: Focusing on Computer Vision and Its Applications*. porquest. URL <https://www.proquest.com/docview/2532452647>.

- [59] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., y Efros, A. A. (2016). *Context Encoders: Feature Learning by Inpainting*. arXiv. URL <https://arxiv.org/abs/1604.07379>.
- [60] Patrick, L. (2017). *Paper review on generative adversarial network (GAN) part 1*. Medium. URL <https://medium.com/@patrickhk/paper-review-on-generativeadversarial-network-gan-part-1-48597bcc96df>.
- [61] Perarnau, G., Wiejer, J., Raduccanu, B., y Álvarez, J. M. (2016). *Invertible Conditional GANs for image editing*. arXiv. URL <https://arxiv.org/abs/1611.06355>.
- [62] Radford, A., Metz, L., y Chintala, S. (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1511.06434>.
- [63] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., y Lee, H. (2016). *Generative Adversarial Text to Image Synthesis*. arXiv. URL <https://arxiv.org/abs/1605.05396>
- [64] Rosebrock, A. (2019). *Deep Learning for Computer Vision with Python*. Practitioner Bundle, PyImageSearch.
- [65] Rosebrock, A. (2019). *Keras ImageDataGenerator and Data Augmentation*. PyImageSearch. URL <https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation>.
- [66] Roth, K., Lucchi, A., Nowozin, S., y Hofmann, T. (2017). *Stabilizing Training of Generative Adversarial Networks through Regularization*. arXiv. URL <https://arxiv.org/abs/1705.09367>.

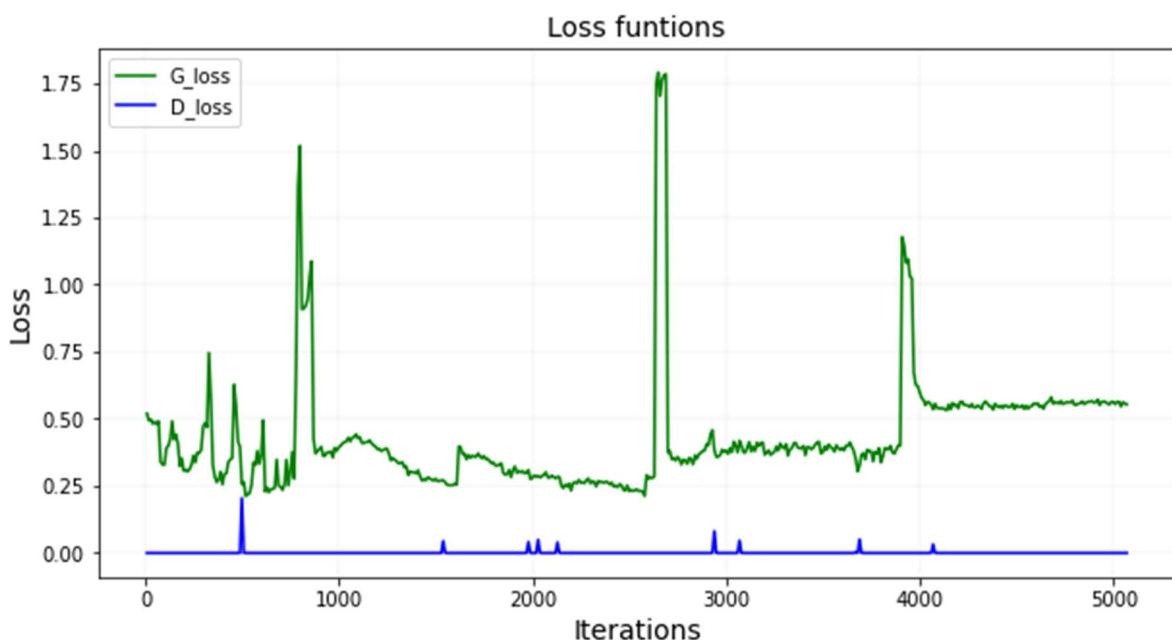
- [67] Rumelhart, D.E., (1986). *Learning internal representations by error propagation*. In *Parallel Distributed Processing. Vol 1: Foundations*. MIT Press, Cambridge, MA.
- [68] Ruthotto, L. y Haber, E. (2021). *An Introduction to Deep Generative Modeling*. arXiv. URL <https://arxiv.org/abs/2103.05180>.
- [69] Sajjadi, M. (2016). *Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning*. arXiv. URL <https://arxiv.org/abs/1606.04586>.
- [70] Shapiro, L., Stockman, y G. *Computer Vision*. (2001). Pearson Education, Inc.
- [71] Shorten, C. (2019). StyleGAN2. Towards data science. URL <https://towardsdatascience.com/stylegan2-ace6d3da405d>.
- [72] Sønderby, C. K., Caballero, L., Thesis, L., Shi, W., y Huszár, F. (2016). *Amortised MAP Inference for Image Super-resolution*. arXiv. URL <https://arxiv.org/abs/1610.04490>.
- [73] Stanford Vision Lab, Stanford University y Princeton University. (2021). ImageNet. URL <https://image-net.org>.
- [74] Szeliski, R. (2019). Computer Vision: Algorithms and Applications. Texts in Computer Science. Springer London Dordrecht Heidelberg New York.
- [75] Vondrick, C., Pirsiavash, H., y Torralba, A. (2016). *Generating Videos with Scene Dynamics*. arXiv. URL <https://arxiv.org/abs/1609.02612>.
- [76] Wang, Y. (2020). *Transferring and Learning Representations for Image Generation and Translation*. Universitat Autònoma de Barcelona. URL <https://www.tesisenred.net/bitstream/handle/10803/669579/yawa1de1.pdf>.

- [77] Wang, Y., Girschick, R., Hebert, H., y Hariharan, B. (2018). *Low-Shot Learning from Imaginary Data*. arXiv. URL <https://arxiv.org/abs/1801.05401>.
- [78] Wang, Y. González-García, A., Berga, D., Herranz, L., Khan, F. S., y Weijer, J.. (2019). *MineGAN: effective knowledge transfer from GANs to target domains with few images*. arXiv. URL <https://arxiv.org/abs/1912.05270>.
- [79] Wu, G. (s.f.) *IMPLEMENTATION\_Variational-Auto-Encoder*. GitHub. Consultado el 9 de octubre de 2021. URL [https://github.com/wuga214/IMPLEMENTATION\\_Variational-Auto-Encoder](https://github.com/wuga214/IMPLEMENTATION_Variational-Auto-Encoder).
- [80] Wu, J., Zhang, C., Xue, T., Freeman, W. T., y Tenenbaum, J. B. (2016). *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*. arXiv. URL <https://arxiv.org/abs/1610.07584>.
- [81] Xie, Q., Dai, Z., Hovy, E., Luong, M., y Le, Q. V. (2019). *Unsupervised Data Augmentation for Consistency Training*. arXiv. URL <https://arxiv.org/abs/1904.12848>.
- [82] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., y Xiao, J. (2015). *LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop*. URL <https://www.yf.io/p/lsun>.
- [83] Zhai, X., Oliver, A., Kolesnikov, A., y Beyer, L. (2019). *S4L: Self-Supervised Semi-Supervised Learning*. arXiv. URL <https://arxiv.org/abs/1905.03670>.
- [84] Zhang, H., Goodfellow, I., Metaxas, D., y Odena, A. (2018). *Self-Attention Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1805.08318>.

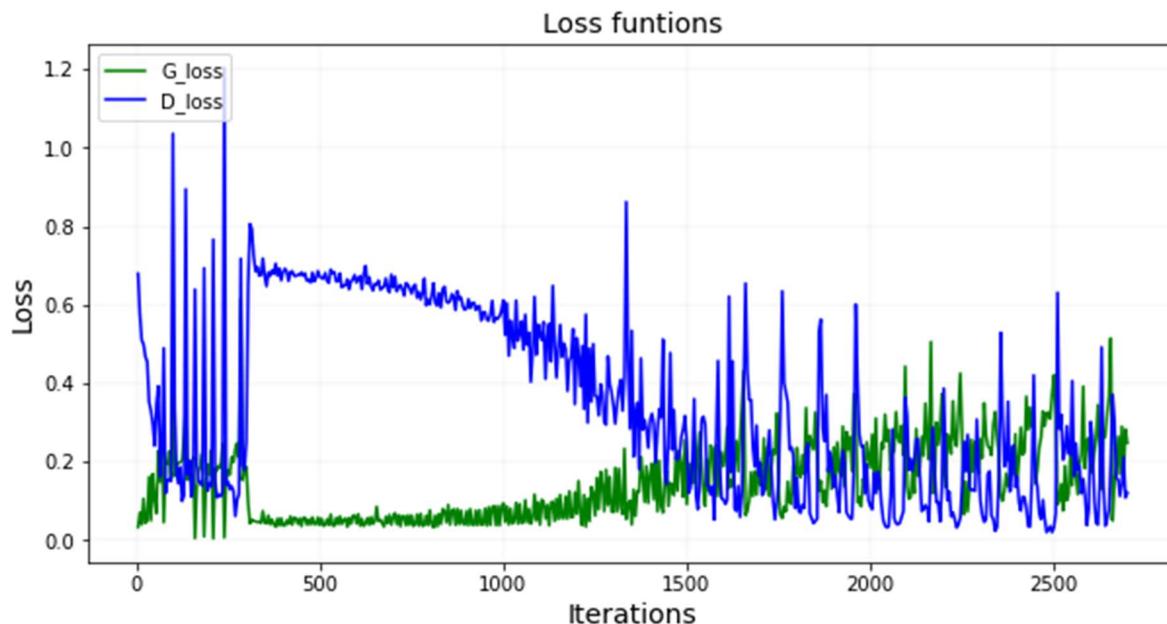
- [85] Zhang, H., Shang, Z., Odena, A., y Lee, H. (2019). *Consistency Regularization for Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1910.12027>.
- [86] Zhang, H., Xu, T., Li, H., Zhang, S., Wnag, X., Huang, X., y Metaxas, D. (2016). *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1612.03242>.
- [87] Zhao, S., Liu, Z., Zhu, J., y Han, S. (2020). *Differentiable Augmentation for Data-Efficient GAN Training*. arXiv, 2020. URL <https://arxiv.org/abs/2006.10738>.
- [88] Zhao, S., Liu, Z., Zhu, J., y Han, S. (2020). *Differentiable Augmentation for Data-Efficient GAN Training. Datasets*. URL <https://data-efficient-gans.mit.edu/datasets>.
- [89] Zhao, Z, Ting Chen, Z. Z., Singh, S., y Zhang, H. (2020). *Image augmentations for GAN training*. arXiv. URL <https://arxiv.org/abs/2006.02595>.
- [90] Zhu, J., Park, T., Isola, P., y Efros, A. (2017). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. arXiv. URL <https://arxiv.org/abs/1703.10593>.

## 9. Anexos

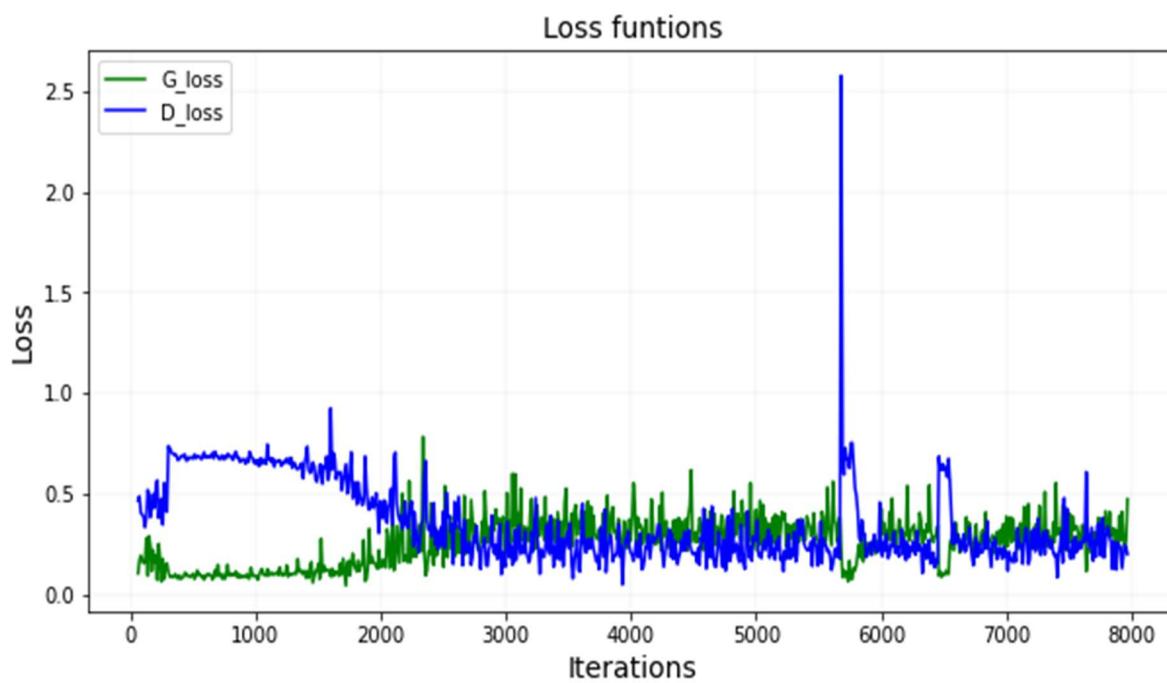
### 9.1. ANEXO 1. GRÁFICAS DE LAS FUNCIONES DE PÉRDIDAS DE LOS EXPERIMENTOS REALIZADOS



Gráfica A1.1: BigGAN - desde cero (8K iter). Dataset 100-Obama.



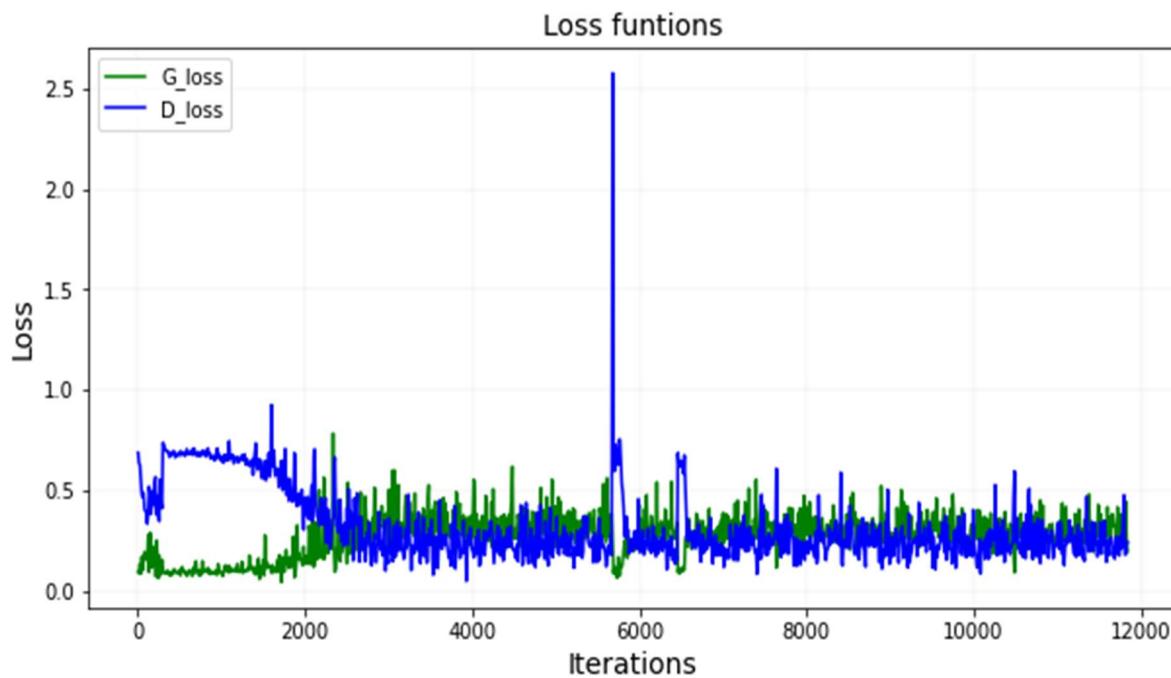
Gráfica A1.2: BigGAN+TC (máx 8K iter). Dataset 100-Obama.



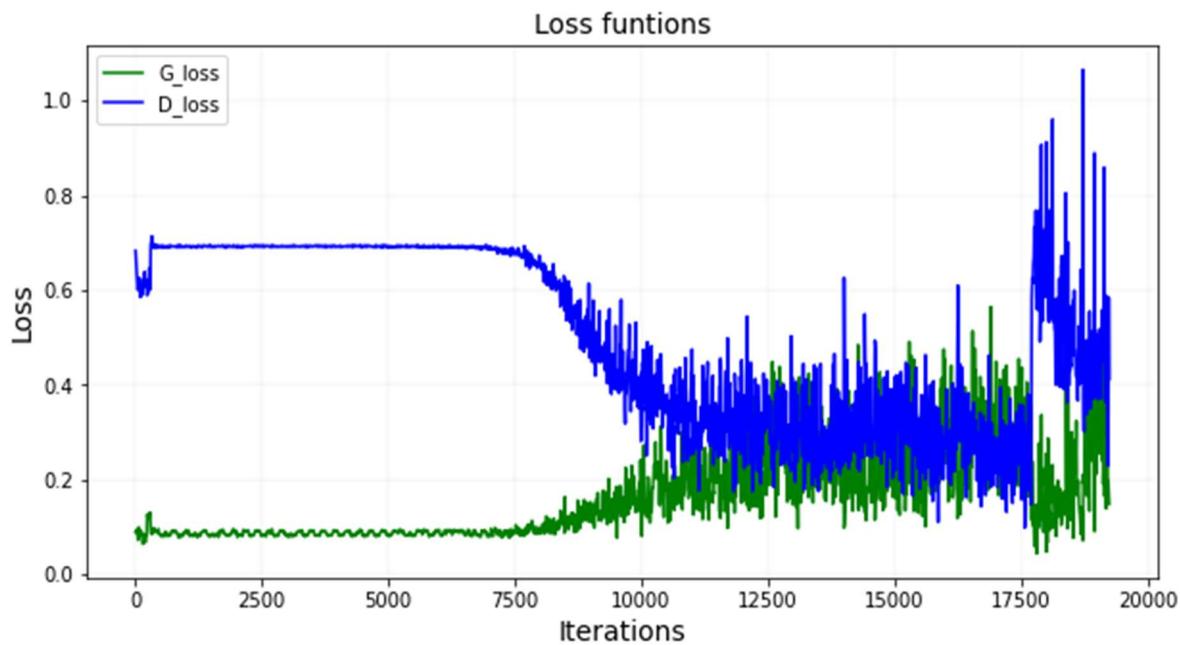
Gráfica A1.3: BigGAN+TC+AD (máx 8K iter). Dataset 100-Obama.



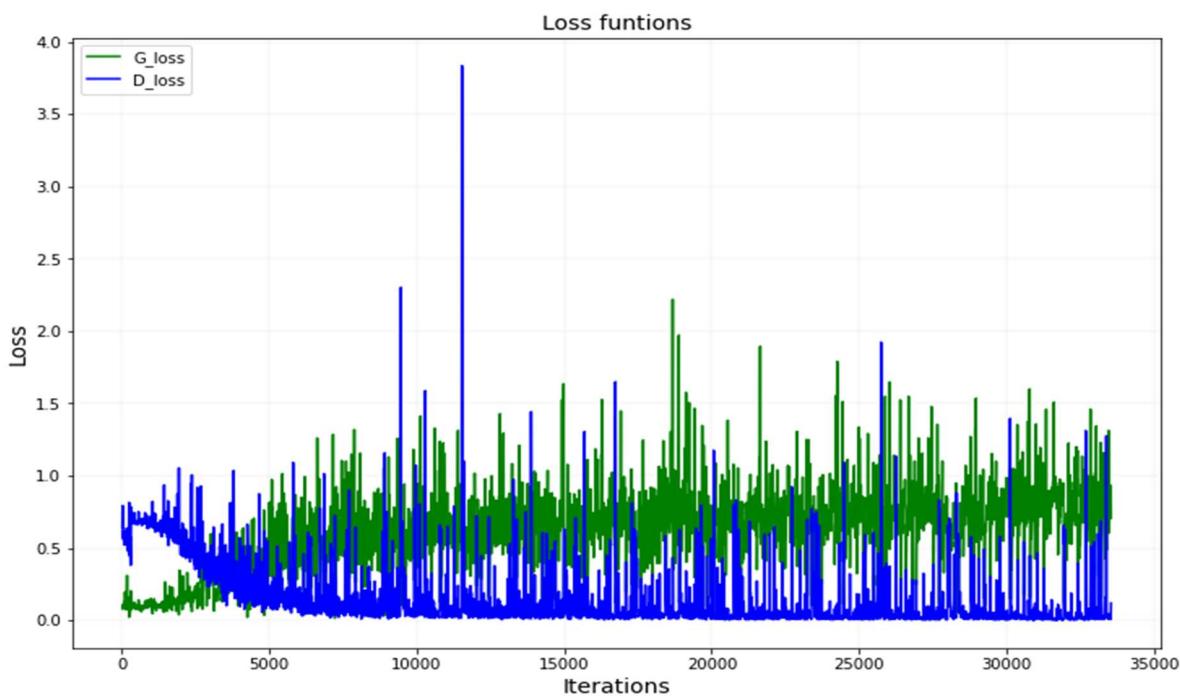
Gráfica A1.4: BigGAN+TC+AD+CR (máx 8K iter). Dataset 100-Obama.



Gráfica A1.5: BigGAN+TC+AD (máx 18K iter). Dataset 100-Obama.

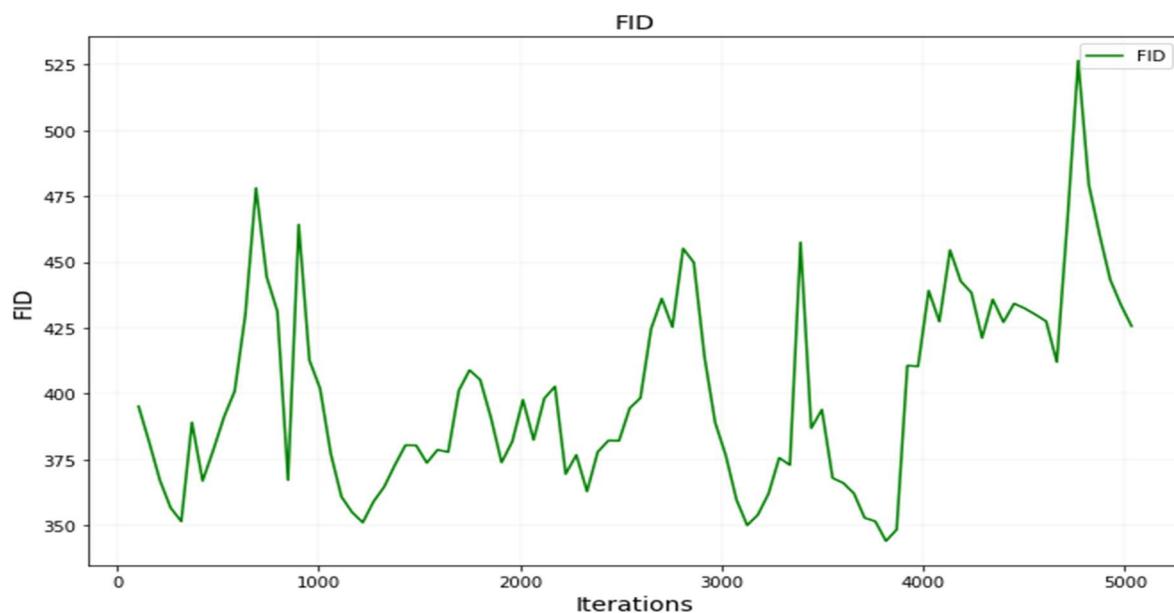


Gráfica A1.6: BigGAN+TC+AD+CR (máx 18K iter). Dataset 100-Obama.

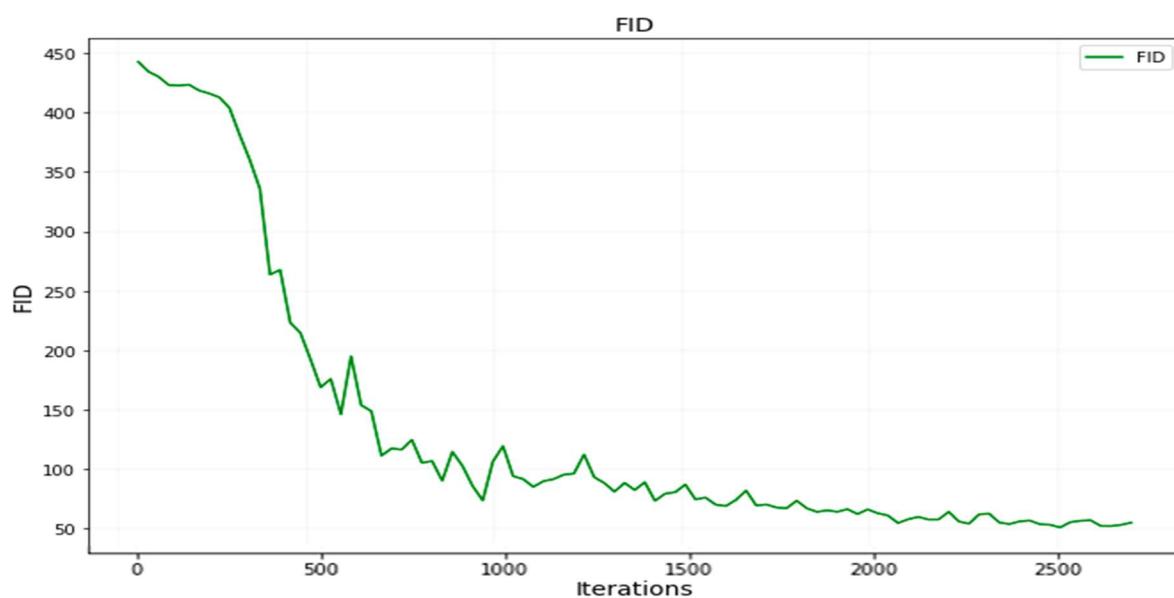


Gráfica A1.7: BigGAN+TC+AD (máx 18K iter). Dataset 1000-LSUN.

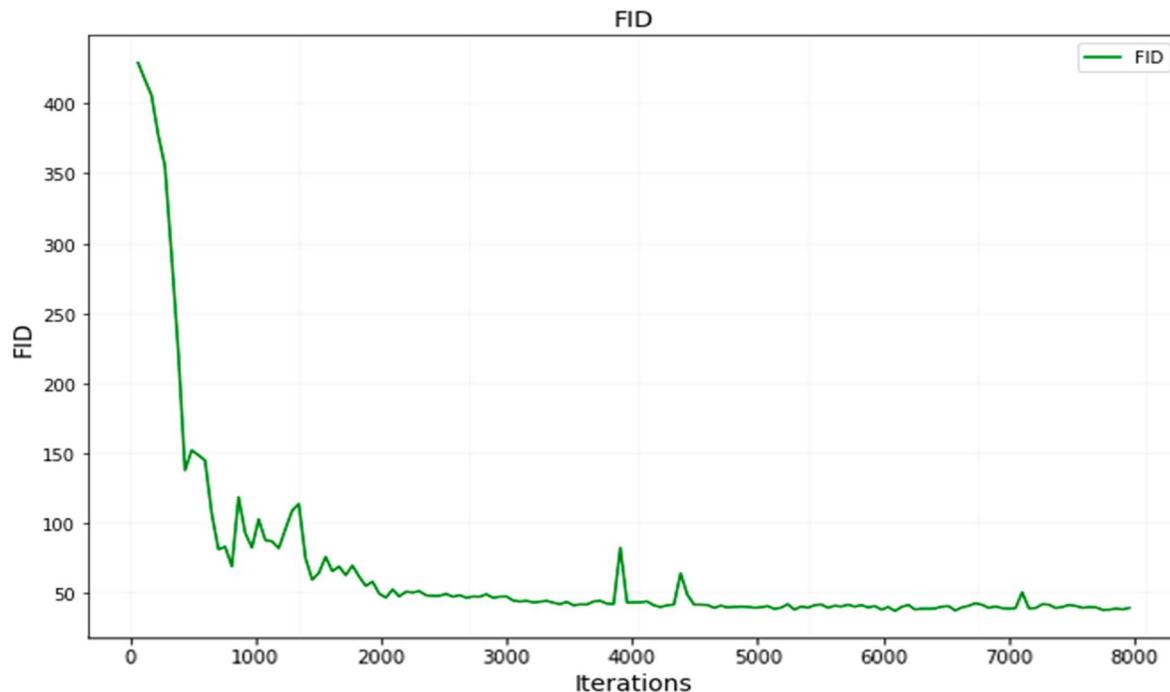
## 9.2. ANEXO 2. GRÁFICAS DEL FID DE LOS EXPERIMENTOS REALIZADOS



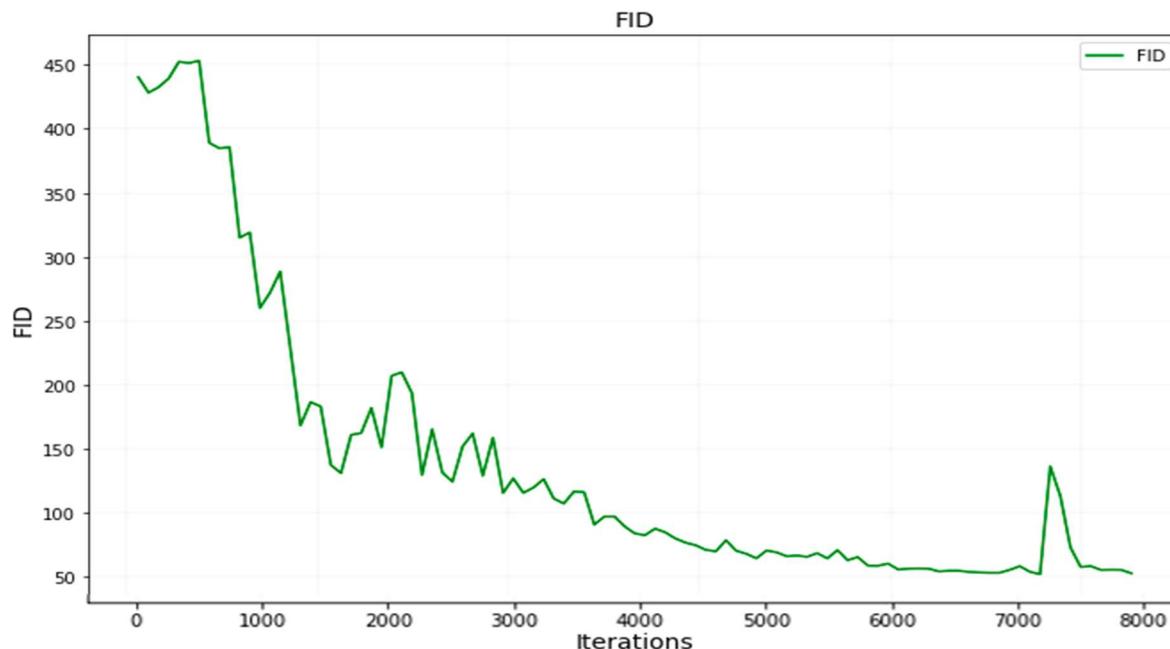
Gráfica A2.1: BigGAN - desde cero (máx 8K iter). Dataset 100-Obama.



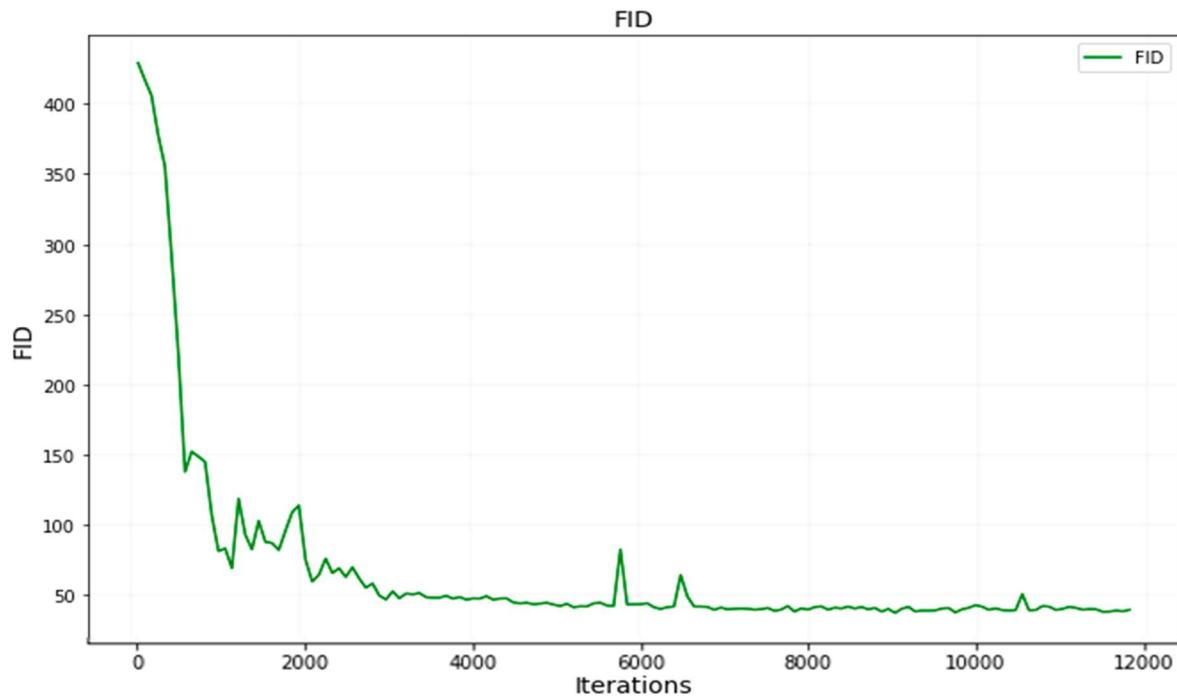
Gráfica A2.2: BigGAN+TC (máx 8K iter). Dataset 100-Obama.



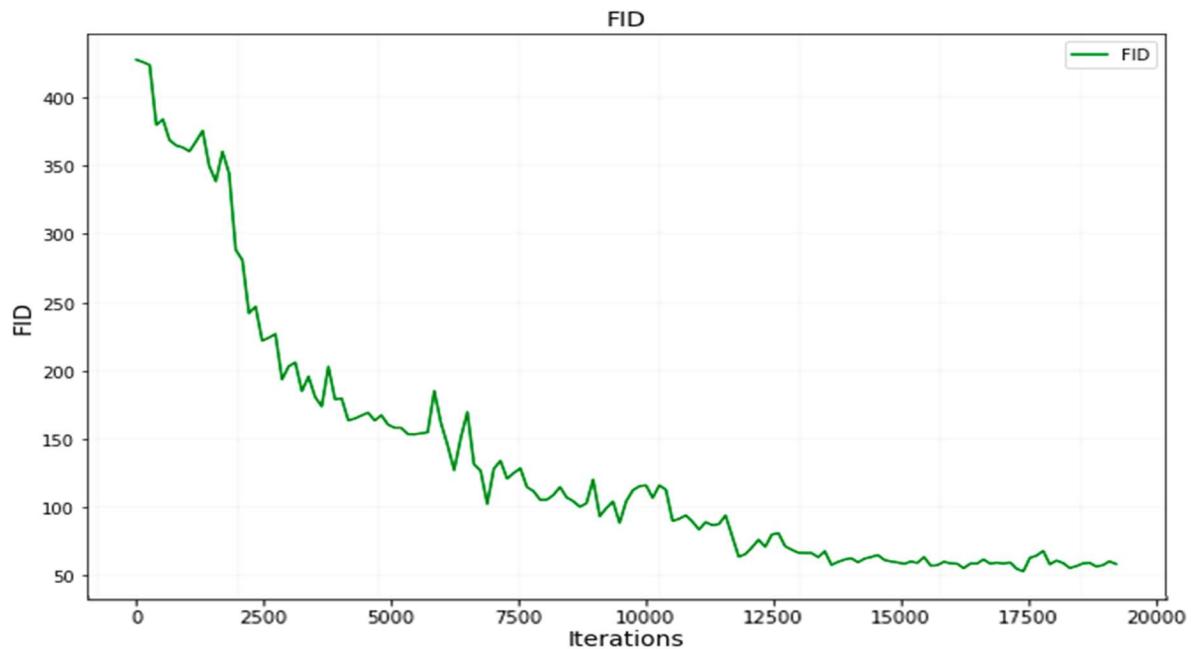
Gráfica A2.3: BigGAN+TC+AD (máx 8K iter). Dataset 100-Obama.



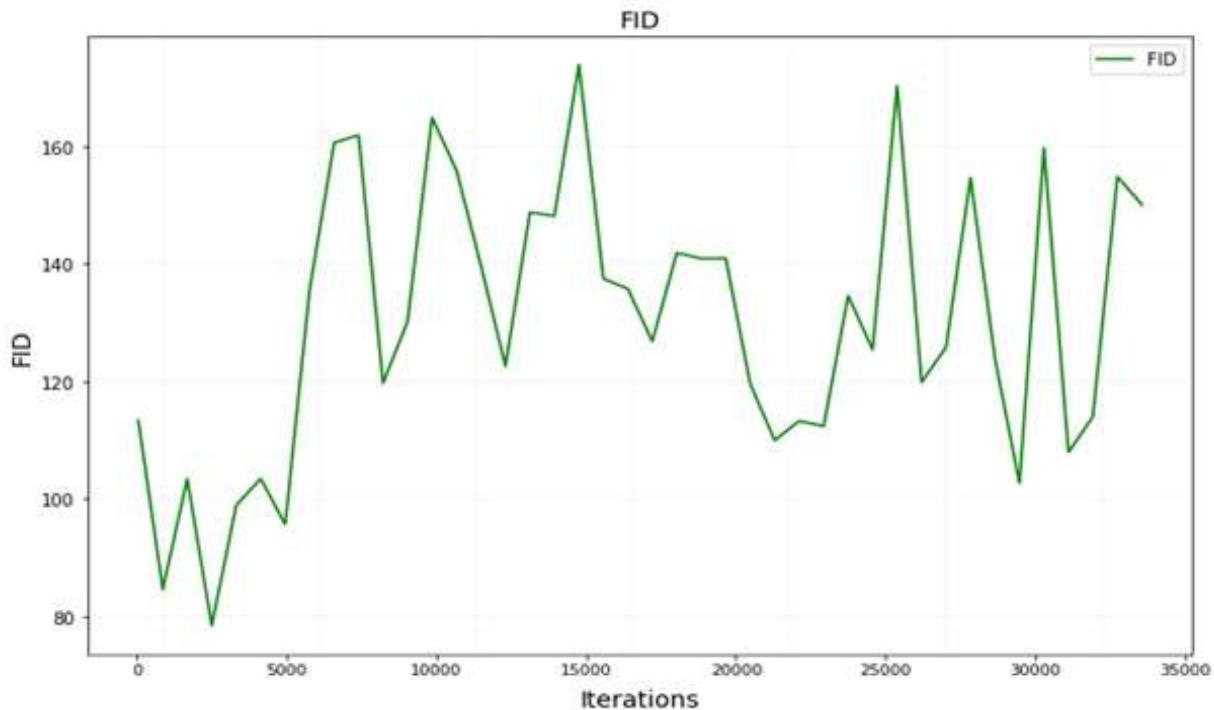
Gráfica A2.4: BigGAN+TC+AD+CR (máx 8K iter). Dataset 100-Obama.



Gráfica A2.5: BigGAN+TC+AD (máx 18K iter). Dataset 100-Obama.



Gráfica A2.6: BigGAN+TC+AD+CR (máx 18K iter). Dataset 100-Obama.



Gráfica A2.7: BigGAN+TC+AD (máx 18K iter). Dataset 1000-LSUN.