Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Inteligencia artificial

Comparativa de algoritmos de generación de data sets sintéticos para vistas aéreas

Trabajo fin de estudio presentado por:	Guillermo Domínguez Muñoz
Tipo de trabajo:	Tipo 3: Comparativa de soluciones
Director/a:	David Guillermo Fernández Herrera
Fecha:	10 de julio 2024

Resumen

Debido al aumento de las aplicaciones de la inteligencia, la demanda de datos para entrenar estos modelos también ha crecido significativamente. Uno de los tipos de datos que se necesitan son las imágenes aéreas, debido a que conseguirlas puede implicar un alto costo, problemas de privacidad o falta de diversidad. Una solución viable es conseguir esas imágenes generándolas sintéticamente a través de modelos de redes profundas. Este proyecto tiene como objetivo comparar varios modelos generadores de imágenes para determinar su capacidad de abordar este problema. Se investigará sobre los diferentes modelos que existen en la actualidad, viendo como se comportan al entrenarlos con otros fines, y seleccionando aquellos que se consideren más prometedores para la generación de imágenes aéreas. Los modelos seleccionados para esta comparativa son FastGAN, ProGAN y StyleGAN3. Para compararlos de manera justa, se establecerá el entorno común en el que se entrenarán los modelos, para que trabajen en las mismas condiciones y así valorar las ventajas y desventajas de cada uno. Se elegirá el sistema de computación, el conjunto de datos de entrenamiento y los métodos de evaluación comunes. Se emplearán métricas cuantitativas, como como el Frechet Inception Distance (FID) y el Inception Score (IS), que proporcionarán valores numéricos con los que se podrá analizar las imágenes generadas con cada modelo, y compararlas con criterios fiables y razonables. Tras entrenar los modelos y examinarlos, se ha llegado a la conclusión de que, entre los modelos seleccionados, StyleGAN3 sería la mejor opción para generar imágenes aéreas, consiguiendo generar imágenes con una calidad y realismo superior, imponiéndose al resto de modelos. Sin embargo, si el tiempo es un factor limitante, con FastGAN se podrían conseguir resultados de buena calidad, en un tiempo muy inferior. En contraste, con ProGAN, ha demostrado un rendimiento inferior en todos los aspectos evaluados.

Palabras clave: Imágenes sintéticas, imágenes aéreas, FastGAN, ProGAN, StyleGAN 3.

Abstract

Due to the increase in AI applications, the demand for data to train these models has also grown significantly. One type of data that is needed is aerial images, as obtaining them can involve high costs, privacy issues, or a lack of diversity. A viable solution is to generate these images synthetically through deep learning models. This project aims to compare various image generation models to determine their ability to address this problem. We will investigate the different models currently available, observing their performance when trained for other purposes, and selecting those deemed most promising for generating aerial images. The models chosen for this comparison are FastGAN, ProGAN, and StyleGAN3. To compare them fairly, we will establish a common environment in which the models will be trained, ensuring they operate under the same conditions to assess the advantages and disadvantages of each. The computing system, training dataset, and common evaluation methods will be selected. Quantitative metrics such as the Frechet Inception Distance (FID) and the Inception Score (IS) will be employed to provide numerical values that allow for the analysis of the images generated by each model, comparing them with reliable and reasonable criteria. After training and examining the models, it has been concluded that among the selected models, StyleGAN3 would be the best option for generating aerial images, achieving superior quality and realism, outperforming the other models. However, if time is a limiting factor, FastGAN could produce good quality results in a much shorter time. In contrast, ProGAN demonstrated inferior performance in all evaluated aspects.

Keywords: Synthetic images, aerial images, FastGAN, ProGAN, StyleGAN 3.

Índice de contenidos

1. Introducción 1
1.1. Motivación 1
1.2. Estructura del trabajo 2
2. Objetivos concretos y metodología de trabajo 4
2.1. Objetivo general 4
2.2. Objetivos específicos 4
2.3. Metodología del trabajo 5
3. Contexto y estado del arte 7
3.1. Contexto del problema 7
3.1.1. Problema de escasez de Datasets 7
3.1.2. Data sets sintéticos 8
3.1.3. Data sets Aéreos 10
3.2. Estado del arte 13
3.2.1. Generación manual 13
3.2.2. GAN 15
3.2.3. VAE 18
3.2.4. VQ-VAE 19
3.2.5. DCGAN 20
3.2.6. QGAN 21
3.2.7. StyleGAN 21
3.2.8. StyleGAN3 23
3.2.9. ProGAN 24
3.2.10. FastGAN 25
3.3. Conclusiones del estado del arte 27

4. Planteamiento de la comparativa 28
4.1. Plataformas en la Nube para el Entrenamiento de Modelos 28
4.1.1. Microsoft Azure 29
4.1.2. Google Cloud Platform (GCP) 30
4.1.3. Amazon Web Services (AWS) 31
4.1.4. Nvidia 32
4.1.5. IBM 33
4.1.6. Problema con las plataformas en la nube 34
4.2. Entorno del entrenamiento local 35
4.3. Dataset 35
4.4. Métricas 37
4.4.1. Evaluación cualitativa 38
4.4.2. Métricas cuantitativas 39
4.4.2.1. Frechet Inception Distance (FID) 40
4.4.2.2. Inception Score (IS) 41
4.4.2.3. Kernel Inception Distance (KID) 42
4.4.2.4. Geometric score 43
4.5. Parámetros del entrenamiento 44
4.5.1. FastGAN 47
4.5.2. ProGAN 48
4.5.3. StyleGAN3 49
4.6. Criterios de éxito de la comparativa 50
5. Desarrollo de la comparativa 51
5.1. Imágenes Generadas 51
5.2. Aplicación de las métricas 52

6. Discusión y Análisis de los resultados 55

7. Conclusiones y trabajo futuro 59

7.1. Conclusiones 59

7.2. Líneas de trabajo futuro 61

Referencias bibliográficas 63

Anexo A. Ejemplos de imágenes del dataset 71

Anexo B. Ejemplos de imágenes generadas 72

Anexo C. Código empleado 75

Índice de figuras

Figura 1 Imágenes generadas de manera artificial. Caras de diferentes ropas expresiones, poses e iluminación. Fuente: Wood, E. et al., 2021 9

Figura 2 Ejemplos de mapas generados con GeoGAN Fuente: Ganguli, S. et al., 2019 11

Figura 3 Ejemplo de Cloud-GAN Fuente: Singh, P., y Komodakis, N., 2018 12

Figura 4 Renderización con UnrealEngine4 de la misma escena bajo diferentes condiciones atmosféricas, (a) soleado, (b) nublado, (c) tarde y (d) neblina. Fuente: Gao, Q. (2020) 13

Figura 5 Imágenes sintéticas de ojos Fuente: Świrski, L. y Dodgson, N., 2014 14

Figura 6 Ejemplos de imágenes generadas sintéticamente Fuente: Rampini, L. y Re Cecconi, F. 2024 14

Figura 7 Ecuación de la función minimax de GAN 16

Figura 8 Ejemplo de arquitectura GAN Fuente: Little, C et al., 2024 16

Figura 9 Arquitectura VAE Fuente: Huang, H et al., 2018 18

Figura 10 Arquitectura VQ-VAE. Fuente: Razavi, A. et al., 2019 19

Figura 11 Arquitectura de la DCGAN Fuente: Zhao, S. et al., 2020 20

Figura 12 Generador Style-based Fuente: Karras, T. et al., 2019 22

Figura 13 Arquitectura del generador del alias-free StyleGAN3. Fuente: Karras, T. et al., 2021 23

Figura 14 Arquitectura ProGAN Fuente: Karras, T. et al., 2017 25

Figura 15 Estructura del módulo Skip-layer y del Generador de una FastGAN. Fuente: Liu, B. et al., 2020 26

Figura 16 La estructura y el flujo hacia adelante del Discriminador. Fuente: Liu, B. et al., 2020 26

Figura 17 Ejemplos de las categorías en el conjunto de datos AID Fuente: Xia, G. et al., 2017 36

Figura 18 Ejemplos de imágenes del dataset de Forest Aerial Fuente: Demir, I. et al., 2018 .37

Figura 19 Ecuación para calcular la distancia de Férchet 41

Figura 20 Ecuación para el cálculo de IS 41

Figura 21 Ecuación de la distribución marginal 42

Figura 22 Ecuación de máxima discrepancia media aplicada a imágenes 43

Figura 23 Ejemplo imágenes generadas con FastGAN 51

Figura 24 Ejemplo imágenes generadas con ProGAN 51

Figura 25 Ejemplo imágenes generadas con StyleGAN3 52

Figura 26 Gráfico con valores de MRLT de las imágenes reales y generadas de ProGAN 53

Figura 27 Gráfico con valores de MRLT de las imágenes reales y generadas de StyleGAN 53

Figura 28 Gráfico con valores de MRLT de las imágenes reales y generadas de FastGAN 54

Figura 29 Imágenes aleatorias del dataset Fuente: Demir, I. et al., 2018 71

Figura 30 Imágenes aleatorias de generadas con FastGAN 72

Figura 31 Imágenes aleatorias de generadas con ProGAN 73

Figura 32 Imágenes aleatorias de generadas con Stylegan3 74

Índice de tablas

Tabla 1 Ventajas y desventajas de Azure Fuente: Choudhary, A et al., 2022 29

Tabla 2 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022 30

Tabla 3 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022 32

Tabla 4 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022 33

Tabla 5 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022 34

Tabla 6 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 6 para las distintas resoluciones de imagen 45

Tabla 7 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 8 para las distintas resoluciones de imagen 45

Tabla 8 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 12 para las distintas resoluciones de imagen 46

Tabla 9 Valores de los parámetros del entrenamiento de FastGAN 48

Tabla 10 Valores de los parámetros del entrenamiento de ProGAN 49

Tabla 11 Valores de los parámetros del entrenamiento de StyleGAN3 50

Tabla 12 Resultados de las métricas 52

Tabla 13 Tiempo de entrenamiento de cada modelo 54

1. Introducción

Disponer de conjuntos de imágenes que cumplan los requisitos de calidad y diversidad que solicitan las tecnologías y avances de hoy es un problema que es difícil de resolver en muchos aspectos. Generar imágenes de manera sintética es una solución posible, dado el increíble avance que están teniendo las inteligencias artificiales. Un ámbito que se podría ver beneficiado por esta solución es la generación de datasets de imágenes aéreas, los cuales pueden requerir de un alto coste, debido que suelen necesitar equipos especializados, como drones o satélites, para poder obtenerlas. Además, capturar este tipo de imágenes implica la recopilación de datos sensibles o privados, al captar propiedades privadas, personas o actividades sin consentimiento, puede generar problemas éticos y legales sobre privacidad. También asegurar la diversidad de los datasets es crucial, y en imágenes aéreas es complicado conseguir imágenes representativas de las diferentes regiones, con diferentes condiciones climáticas, por lo que pueden estar sesgados hacia fotos de áreas más accesibles. Por ello, en este proyecto se realizará una comparativa de algunos modelos generadores de imágenes, para ver su capacidad de generar este tipo de imágenes, ver cómo se comportan y cuales podrían ser una solución posible para generar conjuntos de imágenes aéreas.

1.1. Motivación

La inteligencia artificial (IA) se encuentra en un momento de desarrollo e innovación en prácticamente todos los aspectos en los que se pueda aplicar. Para poder realizar las mejores investigaciones y emplear de la mejor manera el ML (Machine learning) se requiere una alta cantidad de datos que cuenten con la calidad suficiente, que represente toda la diversidad posible dentro de la toda la variedad de datos, además de contar con un etiquetado correcto que los clasifique de manera correcta y facilite su uso. En muchos ámbitos esto puedo resultar un verdadero reto, llegando a ser incluso el factor limitante que frene el avance de algún proyecto. Puede ser debido a diferentes motivos como el coste para adquirirlos, limitaciones legales y políticas, que tengan calidad insuficiente, problemas de privacidad, etc.

La disponibilidad de un buen data set de imágenes aéreo es una de esas áreas donde puede haber problemas para conseguir un data set de buena calidad y diversidad. Para

conseguirlas requiere de drones, aviones o satélites que implican un alto costo, además de que puede generar problemas de privacidad, permisos y regulaciones, y teniendo en cuenta también las condiciones climáticas que pueden afectar tanto a la calidad como a la variedad. Un data set de este tipo tendría numerosas aplicaciones en cartografía para la generación y el manejo de planos, planificación urbana y agrícola, desarrollo de infraestructuras, para controlar y gestionar los recursos y los desastres naturales, fines militares, de seguridad y vigilancia, y muchos otros.

Para solucionar todos estos problemas una solución puede ser la propia IA mediante la generación de datos sintéticos, datos creados de manera artificial, que podrían tener un valor equivalente al de los datos reales si cumplen con las características necesarias. Esto añade otra ventaja, la posibilidad de controlar y modificar diversas variables, como la iluminación, las condiciones meteorológicas o la perspectiva de las imágenes, permitiendo crear conjuntos de datos más ricos, evitando limitaciones físicas y logísticas que sí pueden tener las imágenes reales.

Por lo tanto, la generación de imágenes sintéticas es una solución novedosa y eficaz a los problemas actuales que han surgido con la necesidad de conjuntos de datos aéreos. Este TFM se propone investigar y avanzar en este ámbito, aportando al desarrollo de una inteligencia artificial más sólidas y adaptables a las distintas áreas. La investigación no solo tiene el potencial de mejorar la precisión y la eficiencia de los modelos de visión por computadora, al facilitar el acceso a datos de calidad con mejor diversidad, sino también de abrir nuevas oportunidades para el uso de la IA en el análisis de imágenes aéreas, beneficiando a múltiples sectores y promoviendo el avance tecnológico en este campo emergente.

1.2. Estructura del trabajo

Este trabajo se podrá dividir en cinco capítulos principales que se pueden resumir de la siguiente manera:

- Primero se ha realizado la introducción del proyecto.
- Segundo es el estudio previo: se realizará una investigación sobre la causa del problema y cómo se pretende solucionar junto con la aportación que se realizará en

este proyecto. Además, se buscarán y estudiarán el estado del arte de los generadores de imágenes donde se seleccionarán cuáles se van a comparar.

- Tercero se establecen objetivos: se expondrán los objetivos, tanto el general como los específicos, que se pretenden lograr en este proyecto, y también se indicará la metodología que se seguirá.
- Cuarto se encuentra el desarrollo de la comparativa: se realizará el entrenamiento de los modelos y se expondrán los objetivos que se pretenden lograr en el proyecto. Después, se determinarán las condiciones de entrenamiento, junto con el dataset a utilizar y las métricas que se utilizarán para su evaluación. Se recogerán los resultados obtenidos y se aplicarán las métricas para realizar el análisis y la evaluación de los modelos para compararlos.
- Quinto es el análisis y conclusiones: Después se sacarán las conclusiones y se desarrollarán las deducciones conseguidas, junto con la evaluación de los objetivos propuestas. Además, se expondrán las líneas de trabajo futuro.

2. Objetivos concretos y metodología de trabajo

En este apartado se plasmarán el objetivo general del proyecto, los objetivos específicos en los que se divide, para que sea más sencillo analizar el resultado final, y la metodología de trabajo que se seguirá a lo largo de todo el trabajo.

2.1. Objetivo general

En este proyecto el objetivo general es realizar una comparativa entre diferentes métodos de generación de imágenes, a través de inteligencia artificial, con el fin de investigar cómo se comportan y cuál serían la mejor opción para crear un conjunto de fotos aéreas. Se realizaría entrenando los modelos en condiciones controladas y evaluando los resultados de las imágenes que se produzcan.

2.2. Objetivos específicos

Para lograr el objetivo general, se establecen los siguientes objetivos específicos:

- Identificar cuál es el nivel de relevancia que supone el aumentar la información sobre la generación de imágenes sintéticas, y más concretamente, sobre la generación de imágenes aéreas.
- Investigar sobre los distintos métodos de generación de imágenes existentes, para poder seleccionar cuáles serían los mejores para poder realizar esta comparativa.
- Decidir y preparar cuál será el entorno de entrenamiento controlado que permita ejecutar los diferentes modelos de una manera controlada y que permita cumplir los plazos de entrega.
- Seleccionar un dataset de imágenes aéreas para emplearlo en el entrenamiento de los modelos, que sirva para representar a este tipo de datasets de manera genérica, permitiendo que los resultados de la comparativa se puedan generalizar para todas las imágenes aéreas.
- Establecer cuáles serán las métricas o los métodos de evaluación que permitirán evaluar y comparar los diferentes modelos.
- Analizar los resultados con las imágenes generadas, aplicando los métodos de evaluación que se hayan establecido, para poder compararlos de una manera justa y objetiva.

2.3. Metodología del trabajo

Para cumplir el objetivo de este proyecto, al tratarse de una comparación de varios modelos se seguirá una estructura básica de una comparativa: investigación, desarrollo, y conclusiones. Consistirá en seleccionar unos modelos que puedan procesar imágenes y entrenarlos en el mismo entorno con las mismas condiciones para analizar su comportamiento y compararlos entre sí para valorar los pros y contras de cada uno de ellos en la generación de imágenes aéreas.

La investigación se centrará en determinar que modelos serán los que se seleccionen para compararlo. La información se recogerá a través de internet buscando principalmente artículos científicos donde se expongan modelos creados y aplicados para diferentes usos. Se elegirán tres modelos diferente, basándose en los comportamientos que hayan tenido al ser aplicado para la generación de imágenes, y que por la información que se recoja de ellos se determine que puede tener valor ponerlos a prueba en la generación de imágenes aéreas. Además de esto, también se investigará sobre la situación actual de los conjuntos de imágenes, y en concreto de las imágenes aéreas, para poner en contexto el problema y conocer mejor las ventajas de estudiar este tipo de inteligencias artificiales.

Durante el desarrollo de la comparativa se realizará la comparación de los modelos, pero previamente se determinarán las condiciones de sus entrenamientos, el dataset que se utilizará y las métricas que se emplearán. Se decidirá la mejor plataforma de entrenamiento que se pueda disponer. Se realizará una investigación comparando las plataformas de procesamiento en la nube que proporcionan este servicio y se determinará cual cumple con las condiciones optimas para realizar el entrenamiento de los modelos. En caso de no encontrar ninguna que pueda proporcionar las características requeridas, se empleará un ordenador de manera local para el entrenamiento. Para seleccionar el dataset que se empleará en el entrenamiento de todos los modelos, se buscará entre los que se pueda disponer para seleccionar el que se adapte mejor a la capacidad computacional que se disponga, seleccionando un tamaño de imagen y un número de imágenes que no sea muy elevado para no aumentar la complejidad del entrenamiento, ni muy pequeño, para poder seguir observado imágenes coherentes. El dataset también se elegirá valorando que

represente de manera generalizada las imágenes aéreas y tener diversidad de entornos. Luego, se seleccionarán las métricas que se emplearán para analizar los modelos. Deberán ser las métricas que mejor permitan determinar la calidad de las imágenes. Se buscarán las métricas más utilizadas y que mejor se hayan comportado evaluando modelos en otras situaciones. Según los valores obtenido con ellas se evaluarán y compararán los modelos. A continuación, se trabajará en que parámetros de entrenamiento se establecerán en cada modelo. Esto será independiente de un modelo a otro, ya que cada modelo tiene los suyos propios y no es posible ponerlos en igualdad de condiciones. Se establecerán intentando conseguir la mayor eficiencia posible en cada uno de ellos, para que el factor común sea que todos se están comportando en su estado optimo en el mismo entorno de entrenamiento. Para lograrlo se harán pruebas variando los parámetros y observando su comportamiento.

En el apartado final se analizarán los resultados y se establecerán las conclusiones a las que se haya llegado. Se recogerán los resultados de la aplicación de las métricas, indicando los sus valores en tablas que faciliten su análisis, junto con el tiempo de entrenamiento, para tenerlo en consideración, y se expondrán de manera representativa las imágenes conseguidas para poder realizar el análisis visual de las mismas. Con toda esta información se realizará un análisis exhaustivo comparando los modelos exponiendo sus ventajas y desventajas para realizar una valoración entre ellos. También, se determinará si se han cumplido los objetivos establecidos, valorándolos uno por uno para establecer si ha el proyecto ha sido un éxito o no. Los razonamientos que se expondrán en las conclusiones deberán ser valoraciones basadas en argumentos que se apoyen en los resultados obtenidos que garanticen la validez y fiabilidad del análisis y la comparación.

3. Contexto y estado del arte

En este apartado se expondrá el contexto y la relevancia del problema a tratar en este proyecto junto con el estado del arte de los modelos generadores de imágenes actuales.

3.1.Contexto del problema

3.1.1. Problema de escasez de Datasets

En el aprendizaje profundo disponer de una cantidad considerable de datos es un elemento crucial para poder alcanzar un resultado óptimo, por lo que las limitaciones para disponer de grandes conjuntos de datos etiquetados es una de las principales barreras para su avance y progresión efectiva. Además, a veces se puede disponer de ellos, pero la tarea de etiquetar datos suele requerir la intervención de anotadores humanos con experiencia. Esto conlleva altos costos, demanda mucho tiempo y puede resultar en errores e inconsistencias, además de ser propensas al ruido, y tienden a variar entre diferentes conjuntos de datos, lo que dificulta la comparación del rendimiento de los modelos entre distintas bases de datos. Incluso en algunos casos, resulta impracticable realizar anotaciones manuales debido a la complejidad de las imágenes o a limitaciones de recursos. Por ejemplo, en imágenes médicas, el nivel de conocimiento requerido para una correcta anotación puede ser tan alta que solo especialistas muy capacitados pueden realizarla. Esto no solo incrementa el costo, sino que también puede alargar significativamente el tiempo necesario para preparar los datos. Además, la cantidad de especialistas disponibles puede ser limitada, creando un cuello de botella en el proceso. Por ello, situaciones donde las imágenes contienen gran cantidad de detalles o características complejas, la tarea de anotación se vuelve aún más complicada de realizar manualmente de manera eficiente.

La falta de diversidad es otro desafío significativo al que se enfrenta la inteligencia artificial, ya que puede generar sesgos en los modelos resultantes, haciendo que disminuya su capacidad de generalización, lo que podría provocar modelos que discriminen a las partes minoritarias de los datos.

Habría que añadir que en algunos casos pueden surgir problemas de privacidad con los datos requeridos. En ámbitos como la medicina, el financiero u otros en los que se utilicen datos personales, se debe tener especial cuidado debido a la sensibilidad de estos temas. Dado el

incremento en el uso masivo de datos, se trabaja para implementar políticas y prácticas que protejan la privacidad de las personas y reduzcan los riegos asociados que pueda causar su uso con la inteligencia artificial (Bellovin, 2019).

Los data sets sintéticos surgen como una solución capaz de abordar todas estas limitaciones de disponibilidad y calidad de los datos. Ofrecen una vía alternativa y controlada para generar y etiquetar imágenes, lo que permite disminuir los desafíos relacionados con la escasez de datos, la falta de diversidad en las muestras y la privacidad. Para lograrlo emplea métodos automáticos para generar las imágenes, con lo que se reduce de manera significativa el tiempo requerido para la obtención de datos y, sobre todo, para el proceso de etiquetado asociado.

3.1.2. Data sets sintéticos

Los datos sintéticos son datos creados de forma artificial, datos virtuales, cuyo propósito es el de simular conjuntos de datos reales. Estos datos se pueden conseguir de una manera rápida y económica, incluyendo etiquetas coherentes y precisas, disminuyendo en los costes de etiquetado. Con estos datos sintéticos se pueden entrenar modelos de procesamiento de imágenes, siendo una valiosa alternativa a los data sets reales, ahorrando costes y problemas legales (Man y Chahl, 2022).

Los datos sintéticos también pueden ayudar a resolver el problema de la falta de diversidad dentro del conjunto de datos, ayudando a disminuir sesgos en los modelos por la falta de representación o el desequilibrio de etiquetas. Aunque hay que tener en cuenta que los datos sintéticos también han sido creados a partir de datos reales de referencia, por lo que también pueden estar sesgados. Para evitar esto es importante analizar correctamente los datos originales, aplicar técnicas de balanceo de datos, como el sobre muestreo de las clases minoritarias, o incluir elementos de diferentes fuentes, entre otras posibles soluciones.

Otro problema que puede surgir al tratar con data sets sintéticos es disparidad entre la distribución de los datos de entrenamiento y los que se utilizan en la evaluación. Esto se conoce como Domain Gap (DG) (Hutchinson et al., 2022), y hace referencia a que en el machine learning se sugiere que las distribuciones de los datos de entrenamiento y evaluación son iguales, lo cual no así siempre. El Domain Gap tiene consecuencias en el modo de comportarse de los modelos siempre que se entrene con un conjunto de datos y se

evalué con otros, pero especialmente en los casos en los que los datos sintéticos se utilicen para el entrenamiento y los reales para la evaluación.

Figura 1 Imágenes generadas de manera artificial. Caras de diferentes ropas expresiones, poses e iluminación. Fuente: Wood, E. et al., 2021

La capacidad computacional puede ser otra limitación a la hora de producir grandes conjuntos de datos. Aunque los data sets sintéticos permitan aumentar la cantidad de datos, controlando su generación y facilitando enormemente su etiquetado, solo hay un número limitado de entidades que disponen de infraestructuras que sean capaces de realizar esa tarea a gran escala.

Dentro de las imágenes sintéticas que se utilizan para entrenar modelos, se pueden diferenciar varios tipos (Man y Chahl 2022):

» Las composiciones sintéticas: hace referencia a imágenes reales que han sido manipulados digitalmente para introducir elementos que no estaban originalmente en los datos de la imagen. Esto incluye la manipulación digital del entorno de la imagen, la introducción de objetos sintéticos en la imagen o la fusión de diferentes imágenes reales en una nueva imagen. Un ejemplo sería el SURREAL (Varol et al., 2017), donde se crean imágenes sintéticas superponiendo objetos o personas sintéticas sobre entornos de fondo reales.

» Datos sintéticos virtuales: se refieren a datos de imagen que son completamente sintetizados, sin contener datos reales directamente, lo que puede aplicarse a una amplia gama de datos de imagen sintética. Se pueden categorizar en tres grupos:

- Las escenas virtuales, son las más simples, suelen utilizar la cantidad mínima de objetos 2D y 3D para crear una escena y capturar datos de imagen sintéticos, como la generación de rostros sintéticos para tareas de reconocimiento facial.
- Los entornos virtuales son un paso más allá de las escenas virtuales y comprenden una construcción virtual completa en 3D de un entorno específico, como podría ser el interior de una casa o un cruce peatonal. De cualquier manera, su objetivo es permitir la captura de datos de imagen desde múltiples perspectivas sin arriesgar la degradación de la calidad de los datos debido a problemas como artefactos de objetos.
- Los mundos virtuales son efectivamente entornos virtuales a una escala más grande. Las escenas fuera de un entorno virtual que pueden haber sido un fondo plano en 2D están completamente construidas con eventos que ocurren más allá de la vista de la cámara virtual. Esto se encuentra más comúnmente en datos virtuales capturados de juegos que tienen entornos preconstruidos a gran escala.

3.1.3. Data sets Aéreos

Aunque los data sets de imágenes aéreas no sean en los que se gastan más recursos de investigación, como podrían ser los de caras o conjuntos de objetos, también puede ser un reto interesante para los diferentes modelos de aprendizaje profundo. Tener imágenes aéreas de la superficie terrestre, ya sea observaciones de terrenos diferentes como agrícolas, volcánicos, de plantaciones forestales, etc. O imágenes aéreas satelitales más amplias que tienen varias aplicaciones muy prácticas, como, por ejemplo: mapeo, cartografía y teledirección, involucrarse en diferentes sectores industriales, seguridad e inteligencia, aplicaciones militares, evaluaciones económicas de las regiones o incluso advertencia de desastres.

Los datos de imágenes aéreas generados han sido principalmente creados con estructuras GAN (Generative Adversarial Networks), y se han empleado en el mapeo de terreno a través de la traducción de imagen a imagen. Algún ejemplo sería el trabajo de Ganguli el al., 2019 donde se consiguieron generar mapas a partir de imágenes satelitales, con una arquitectura, a la que denomino GeoGAN. Este exigía la traducción píxel a píxel, aprendiendo la correspondencia directa entre el píxel de entrada y el píxel de salida, con lo que conseguía más precisión en las características que producía. O el trabajo propuesto por Deng et al.,

2019 donde se desarrolló una red generativa adversaria condicional (CGAN, Conditional Generative Adversarial Network) para sintetizar una vista a nivel del suelo de una ubicación dada una imagen aérea, generando imágenes realistas y representativas a nivel del suelo, utilizando la imagen aérea como información auxiliar.

Figura 2 Ejemplos de mapas generados con GeoGAN Fuente: Ganguli, S. et al., 2019

También se han empleado para generar modelos de superresolución para mejorar y ampliar imágenes satelitales borrosas o de baja resolución como propone Jiang et al., 2019. Otro modelo que implementó imágenes aéreas fotorrealistas fue el propuesto por Singh y Komodakis 2018, al que llamaron Cloud-GAN, que ayuda a resolver el problema de las nubes para estudiar el terreno, permitiendo eliminarlas de las imágenes satelitales. Otro ejemplo sería el que desarrollan Anantrasirichai et al, 2019 entrenando una red CNN de arquitectura AlexNet con datos sintéticos para ayudar al radar InSAR a detectar deformaciones en la superficie con una fuerte relación estadística con la erupción. Como se puede observar, todos estos modelos y muchos otros utilizan imágenes aéreas fotorrealistas de alta resolución, pero se centran en la traducción o modificación de imagen por imagen, no en la investigación en el ámbito de la generación de imágenes sintéticas nuevas.

Como estudia Zhao et al., 2021, el avance en este ámbito ha suscitado preocupación sobre la emergencia de la geografía falsa generada por inteligencia artificial y su potencial para transformar la percepción humana del mundo geográfico.

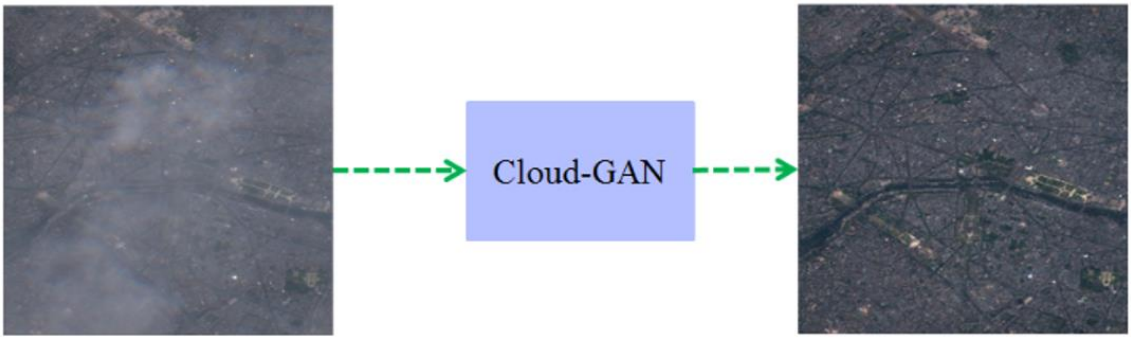

Figura 3 Ejemplo de Cloud-GAN Fuente: Singh, P., y Komodakis, N., 2018

Algunos que sí se han centrado en la generación de imágenes nuevas serían Gao et al, 2020 proponiendo un método de generación de datos sintéticos a gran escala para la comprensión geométrica y semántica de escenas urbanas desde una vista cenital utilizando CityEngine y UnrealEngine4[1]. Alibani et al., 2024 utilizaron StyleGAN 3 para investigar sobre la generación de imágenes satelitales multiespectrales, en particular, intentar conseguir la alta calidad del satélite Sentinel-2. También Audebert et al., 2019 abordan la escasez de datos hiperespectrales anotados necesarios para entrenar redes neuronales profundas y que se puedan implementar en imágenes satelitales.

La generación de imágenes sintéticas aéreas también puede servir como reto para poner a prueba los diferentes métodos de generación de datos sintéticos, como estudian Yates et al., 2022. Aunque la mayoría utilizan caras u objetos como conjunto de imágenes de referencia para entrenar los modelos, el utilizar imágenes de conjuntos de datos más novedosos, como imágenes aéreas, podría suponer cierta ventaja. Esto ocurre debido a que, a diferencia de las imágenes de caras u objetos, las imágenes aéreas presentan una gran variedad de características visuales y patrones, como diferentes tipos de vegetación, superficies con agua, estructuras urbanas y rurales, y cambios en el terreno. Esta diversidad y complejidad pueden hacer que los modelos de generación de datos sintéticos desarrollen una mayor robustez y adaptabilidad. Así, al enfrentarse a una distribución de características visuales

[1] https://www.unrealengine.com/de/spotlights/unreal-studio-brings-cityengine-neighborhood-to-life

más amplia y descentralizada, los modelos no solo se vuelven mejores en generar imágenes aéreas realistas, sino que también adquieren una capacidad mejorada para generalizar a otras tareas.

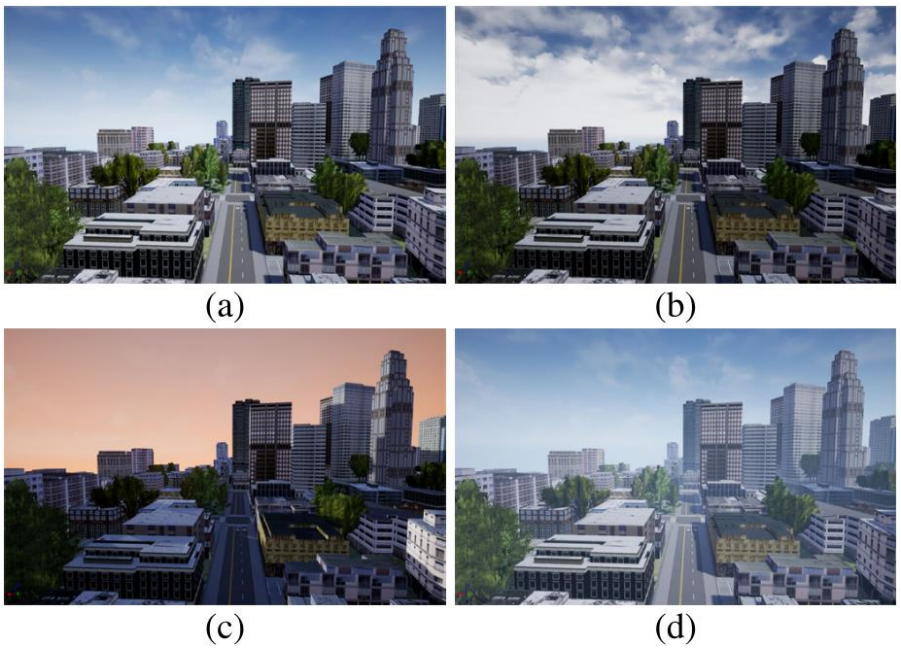

Figura 4 Renderización con UnrealEngine4 de la misma escena bajo diferentes condiciones atmosféricas, (a) soleado, (b) nublado, (c) tarde y (d) neblina. Fuente: Gao, Q. (2020)

3.2. Estado del arte

En este apartado, se presentará el estado del arte en la generación de imágenes sintéticas, dado el objetivo del trabajo, donde se expondrán las técnicas más recientes y efectivas utilizadas.

3.2.1. Generación manual

La producción manual representa la forma más elemental de datos sintéticos, ya sea en forma de imágenes compuestas o entornos tridimensionales, se generan de manera manual, una por una, para conformar un conjunto de datos completo. Este método resulta ser el más exigente en términos de tiempo e, inevitablemente, limita la cantidad de datos que pueden crearse. La tarea de etiquetar y anotar datos sintéticos generados de forma manual suele requerir más trabajo, eliminando los principales beneficios de la síntesis de datos, que se suponía debían ser la generación automática de grandes volúmenes de datos y la anotación

de características de forma automática. Sin embargo, la generación manual de datos sintéticos aún se utiliza en ciertas situaciones (Man y Chahl, 2022).

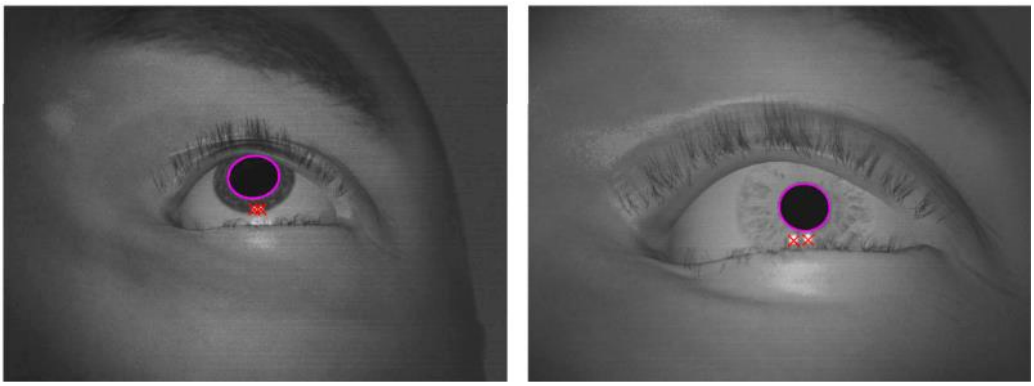

Figura 5 Imágenes sintéticas de ojos Fuente: Świrski, L. y Dodgson, N., 2014

Este método lo utilizaron Swirski y Dodgson, 2014 para hacer un modelo de seguimiento ocular utilizando imágenes sintéticas creadas en Bender. Este es un software de gráficos por computadora en 3D de código abierto que permite tanto el modelado como la renderización de escenas en 3D. Afirman que las imágenes que consiguieron crear son lo suficientemente realistas como para ser utilizadas plausiblemente para la evaluación del procesamiento de imágenes en algoritmos de seguimiento ocular.

Otro ejemplo sería el de Sim2Air (Barisic et al., 2022) un data set sintético creado para monitorizar UAVs. Se creó acentuando la representación de objetos basada en la forma mediante la aplicación de aleatorización de texturas. Se crea un conjunto de datos diverso con foto realismo en todos los parámetros, como forma, pose, iluminación, escala, punto de vista, etc., excepto en texturas atípicas, también utilizando el software de modelado 3D Blender.

Figura 6 Ejemplos de imágenes generadas sintéticamente Fuente: Rampini, L. y Re Cecconi, F. 2024

El artículo de Rampini y Cecconi, 2024 presenta una nueva metodología para generar imágenes sintéticas para la gestión de instalaciones en los edificios. Aprovechando los modelos BIM (Building Information Modeling) en 3D de código abierto y utilizando un motor gráfico, representan espacios interiores fotorrealistas con objetos relacionados con las instalaciones. Con el entorno virtual pueden generar variar la iluminación, posición de la cámara o los materiales, consiguiendo así diferentes imágenes etiquetadas y listas para entrenar modelos.

3.2.2. GAN

Las Redes Generativas Adversarias (GAN) fueron un método propuesto por Goodfellow et al., 2014 y han sido muy utilizadas para la generación de datos sintéticos, tanto imágenes como texto o sonido. Consta de dos redes neuronales que están constantemente compitiendo entre sí en un juego de suma cero, donde la ganancia de una implica la pérdida de la otra y viceversa. Estas redes se denominan generador (G) y discriminador (D).

El generador se encarga de producir imágenes, o cualquier dato, con ruido de manera aleatoria, pero irá aprendiendo con la intención de parecerse lo más posible a la distribución de imágenes o datos reales. La segunda red es un discriminador encargado de distinguir entre los datos reales del entrenamiento y los creados de manera sintética por el generador, clasificándolos como reales o falsos. Por ello no se busca como objetivo minimizar el error, ya que un error en alguna de las redes afecta de manera relativa a la otra. El objetivo real es crear muestras lo más realistas posibles.

Durante la fase de entrenamiento se realiza un proceso antagónico donde las dos redes se entrenan de manera simultánea. El generador se dedicará a mejorar su capacidad de generar imágenes lo más realistas posibles intentando superar al discriminador, y mientras, el discriminador intentará mejorar su capacidad para distinguir entre datos reales y sintéticos, entrando en un juego de minimax. Este juego consta de dos jugadores, en el que uno trata de maximizar sus ganancias mientras que el otro trata de minimizar las ganancias del primero en cada uno de los turnos. Realizando gran cantidad de iteraciones compitiendo entre ellos se consigue mejorar el generador y el discriminador. La siguiente ecuación corresponde a la función error de este tipo de sistema:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

Figura 7 Ecuación de la función minimax de GAN

Donde $\mathbb{E}$ es el operador de valor esperado sobre que valore de *x* se toma de la distribución de datos reales p_{data}, y $\mathbb{E}_{z\sim p_z(z)}$ es lo mismo para que valores de toman de la distribución de ruido. Log D(*x*) es el logaritmo de la probabilidad que el discriminador asigna a que *x* es un dato real. G(*z*) es la muestra generada por el generador usando el ruido *z*, con la que el discriminador D asigna luego una probabilidad para calcular el logaritmo.

La solución a la que lleva esta función es a alcanzar un equilibrio óptimo cuando el generador G reproduzca la distribución real de los datos con un error de 0. Esto significa que debe generar muestras que sean indistinguibles de las reales para el discriminador D, es decir, que la probabilidad de que D pueda identificar el origen de la muestra sea del 50%, hasta alcanzar un punto de equilibrio llamado Equilibrio de Nash. Sin embargo, alcanzar esta solución ideal es complicado en la práctica debido a diversos desafíos que enfrentan las GAN durante el entrenamiento.

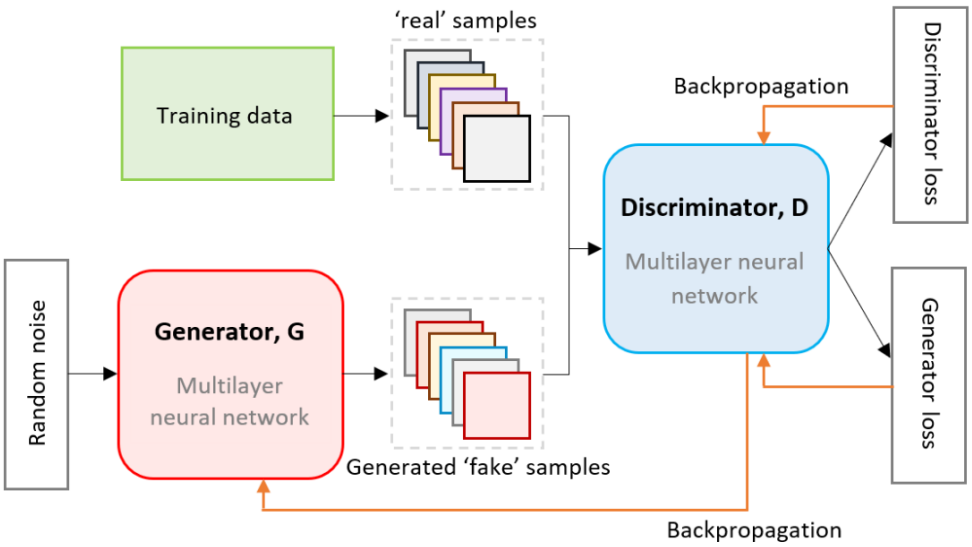

Figura 8 Ejemplo de arquitectura GAN Fuente: Little, C et al., 2024

Los principales problemas con los que se encuentran las redes generativas adversarias son:

- Se denomina "mode collapse" cuando solo se generan imágenes de un solo modo, en otras palabras, las diferentes clases y características con las que cuentan los datos reales no se plasman en ninguno de los datos falsos. Cuando el generador consigue

engañar siempre al discriminador con imágenes iguales o muy similares, provocando que no aprenda nada. Se puede llegar a esta situación si no se tiene en cuenta la diversidad muestral en la manera con la que se trata el error, recompensando la variedad, o penalizando la repetición de imágenes iguales.

- La dificultad para determinar la convergencia del modelo, teniendo en cuenta que incluso existe la posibilidad de que no converja nunca. Como el generador y el discriminador se van optimizando por turnos puede ocurrir que cuando aumenta el error en uno de ellos provoque que disminuya en el otro y viceversa. Esto provoca oscilaciones en el error que influyen en la calidad de los datos generados y generando la incertidumbre al entrenamiento de si el error volverá a disminuir o si ya divergirá de manera indefinida a partir de ese punto. Asegurarse de esto también puedo provocar problemas de coste computacional en el modelo
- Puede ocurrir un desvanecimiento del gradiente. Cuando el entrenamiento ya ha avanzado, el discriminador ha mejorado hasta alcanzar prácticamente su estado ideal, consiguiendo diferenciar entre las imágenes falsa y reales sin equivocarse, provocando que al generador no le llegue suficiente información como para seguir aprendiendo, haciendo que el gradiente que utiliza para entrenar disminuya, ralentizando o anulando su aprendizaje.

Aunque estos problemas pueden complicar mucho el uso de GAN, existen soluciones para resolver o mitigar su impacto (Saxena, D., y Cao, J., 2021):

- Para resolver el “mode collapse” se puede implementar una penalización en la función de pérdida que recompense la diversidad de las imágenes generadas. También se puede emplear un modo enmascarado, introduciendo variaciones en la entrada del generador para forzarlo a producir diferentes modos. O, en lugar de entrenar al generador para engañar directamente al discriminador, entrenarlo para que produzca características similares a las imágenes reales en el discriminador.
- Para resolver el problema de la determinación de la convergencia, vigilando las pérdidas del generador y del discriminador se pueden detectar patrones de oscilación y así poder ajustar las tasas de aprendizaje en consecuencia. Además, se pueden hacer paradas a principio del entrenamiento para evitar el sobre entrenamiento.

- Para resolver el desvanecimiento del gradiente, se puede evitar añadiendo ruido a las etiquetas del discriminador para evitar que se vuelva demasiado seguro. También se pueden utilizar técnicas de batch normalization (Ioffe, S., y Szegedy, C., 2015), normalizando las activaciones de cada capa de la red por cada mini-batch, asegurando una media de cero y varianza de uno, consiguiendo así mejorar la propagación del gradiente.

Este método se ha utilizado en una variedad de aplicaciones, incluida la generación de imágenes, la síntesis de texto, la creación de música, el diseño de productos, la superresolución de imágenes, la generación de caras sintéticas, entre otros.

3.2.3. VAE

Las Variational Autoencoders (VAEs) propuesto por primera vez en Kingma y Welling, 2013, y se emplean en capturar patrones y estructuras complejas en datos de alta dimensión, como pueden ser imágenes, texto o secuencias temporales. Tienen como objetivo aprender representaciones latentes significativas que condensen la información relevante de los datos originales. Al igual que los GAN, están formado por dos redes neuronales llamadas codificador y decodificador.

El codificador es responsable de tomar los datos de entrada y mapearlos a un espacio latente, donde cada punto en este espacio representa una representación codificada de los datos de entrada. Esta representación latente es una codificación compacta que captura las características más importantes de los datos originales. El decodificador se encarga de tomar esa representación de los datos y para reconstruirlo de nuevo, generando datos originales. Cuantos más se entrene, podrá generar datos de mejor calidad ya que aprende mejor los parámetros de la distribución latente.

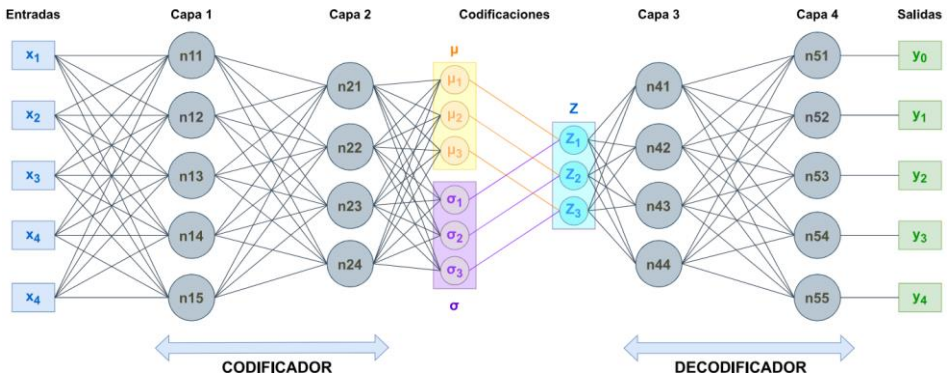

Figura 9 Arquitectura VAE Fuente: Huang, H et al., 2018

Los VAE utilizan técnicas de muestreo estocástico para muestrear puntos del espacio latente durante el entrenamiento y la generación de datos. Esto permite que, a partir de una misma entrada, el modelo genere diferentes instancias de datos, consiguiendo así una generación de datos más diversa y rica. Tienen una capacidad para converger rápidamente y pueden ser útiles para la compresión de datos al aprender representaciones latentes de la información de entrada, sin embargo, su principal desventaja radica en que las imágenes generadas no son de alta calidad, ya que tienden a ser borrosas y carecen de detalles finos.

Se utilizan en una amplia variedad de aplicaciones, incluida la generación de imágenes, la síntesis de texto, la creación de música, la interpolación de vídeos, la imputación de datos perdidos y la compresión de datos, entre otros (Huang et al., 2018).

3.2.4. VQ-VAE

El VQ-VAE es una variante de VAE Van Den Oord y Vinyals, 2017 donde se presenta un nuevo método de entrenamiento, donde se utilizan variables latentes discretas inspiradas en la cuantificación vectorial (VQ). La VQ es una técnica que se utiliza para representar datos continuos mediante una serie de símbolos o códigos discretos. En la VQ-VAE, la distribución posterior y la anterior se modelan como categóricas y se genera una tabla de incrustación (embedding table), que contiene el código discreto, donde se van indexando muestras extraídas de estas distribuciones. Estas incrustaciones servirán de entrada a la red de descodificación. Gracias a este método se consigue aprender de manera más robusta y reconstruir imágenes de mejor calidad (Razavi et al., 2019).

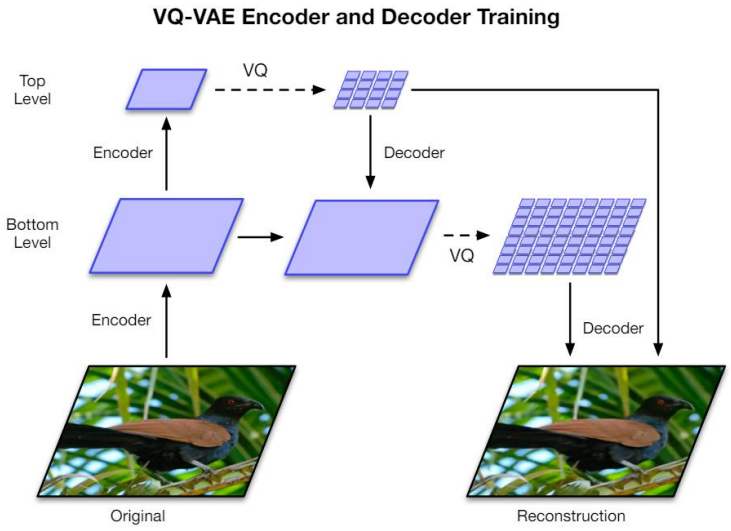

Figura 10 Arquitectura VQ-VAE. Fuente: Razavi, A. et al., 2019

3.2.5. DCGAN

Los DCGAN (Deep Convolutional Generative Adversarial Networks, Gao et al., 2018) son una variante de las GAN creadas específicamente para la generación de imágenes de alta calidad. Sigue utilizando dos redes neuronales, el generador y el discriminador, pero en este caso son CNN (Convolutional Neural Network, redes neuronales convolucionales).

El generador utiliza capas de convolución transpuesta y normalización de lotes para aumentar progresivamente la resolución de las imágenes de salida. Y el discriminador extrae las características de estas imágenes con capas de convolución y las clasifica como reales o falsas.

Durante el entrenamiento se emplea la normalización en lote (Batch Normalization, Ioffe, Sergey, 2015), el cual garantiza que los datos en cada conjunto de datos (o "lote") sigan una distribución estándar con una media de 0 y una varianza de 1. Esto es crucial para prevenir problemas de "explosión del gradiente", donde los valores de los gradientes se hacen tan grandes que causan actualizaciones excesivas en los pesos y variables de la red. Este fenómeno puede generar inestabilidad y dificultar que la red converja hacia una solución óptima.

Se ha comprobado que se pueden mejorar los resultados realizando modificaciones a las imágenes insertadas en el discriminador, tanto a las reales como a las falsas, durante la fase de entrenamiento. Estas modificaciones serían variaciones en el contraste, el brillo, la traslación y la saturación, todo de forma aleatoria (Zhao et al., 2020).

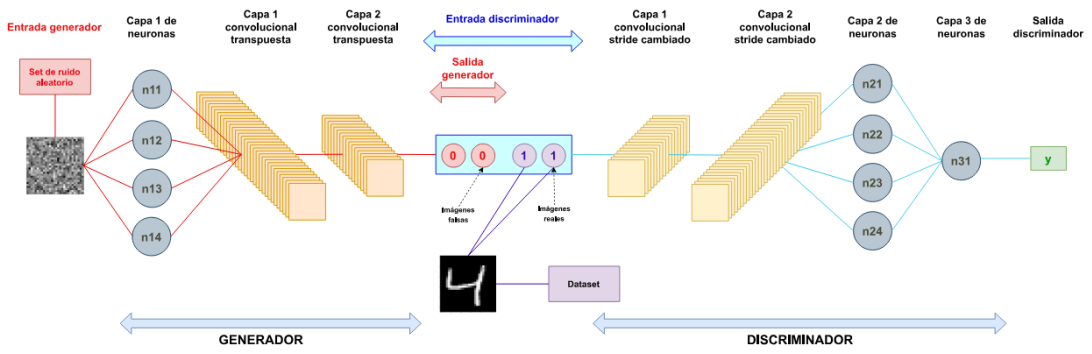

Figura 11 Arquitectura de la DCGAN Fuente: Zhao, S. et al., 2020

3.2.6. QGAN

Dado el progreso en la computación cuántica en los últimos años no sorprende que, por la evolución de los avances tecnológicos, se haya mezclado con el aprendizaje automático, y con las redes generativas adversarias (Bingchen et al., 2021). En las QGAN (Quantum Generative Adversarial Networks) el método de entrenamiento basado en la competición entre adversarios, el generador y el discriminador, permanece igual que en la forma clásica establecida por los GAN. La innovación se encontraría en los métodos subyacentes que se utilizan, donde se aprovechan los principios de la computación cuántica y las capacidades únicas de los circuitos cuánticos. En lugar de entrenar a través de capas de neuronas clásicas, las redes neuronales cuánticas utilizan capas de Anzats (Orellana, 2021) para su proceso de entrenamiento. Los anzats, en este contexto, sirven como circuitos cuánticos parametrizados, ofreciendo un marco flexible y expresivo para tareas de aprendizaje automático cuántico, ofreciendo ventajas potenciales en la generación de datos.

3.2.7. StyleGAN

StyleGAN es una arquitectura de modelo generativo desarrollado por NVIDIA (Karras et al., 2018; Karras et al., 2021) para la síntesis de imágenes fotorrealistas de alta calidad. Introducida por primera vez en 2018, StyleGAN ha sido ampliamente reconocida por su capacidad para generar imágenes convincentes y controlables con una gran variedad de estilos y características visuales.

El generador de esta variante de las GAN propuesto consiste en una serie de bloques de síntesis, cada uno de los cuales se encarga de generar una parte de la imagen final. Cada bloque de síntesis se compone de múltiples capas de convolución y normalización, seguidas de una capa de activación, normalmente ReLU. También la arquitectura del GAN clásico el código latente se introduce a través de una red feedforward, es decir, a través de una capa de entrada. En el StyleGAN, se elimina esta capa de entrada y en su lugar se empieza por una constante aprendida.

Un aspecto destacado de la arquitectura StyleGAN es el uso de la normalización de instancia estilizada (Adaptive Instance Normalization, AdaIN). La AdaIN adapta los parámetros de normalización de instancia en función del estilo de una imagen de referencia, lo que permite

transferir el estilo de una imagen a otra de manera controlada, generando imágenes de manera más precisa

Los bloques de mapeo de estilo toman un vector de ruido como entrada y lo mapean a un espacio de estilo latente. Este espacio de estilo latente codifica la información sobre el estilo de la imagen, incluidos aspectos como la iluminación, el color y la textura. Al modular las operaciones de normalización de instancia en los bloques de síntesis con el espacio de estilo latente, el generador puede adaptar el estilo de las imágenes generadas de acuerdo con el estilo deseado.

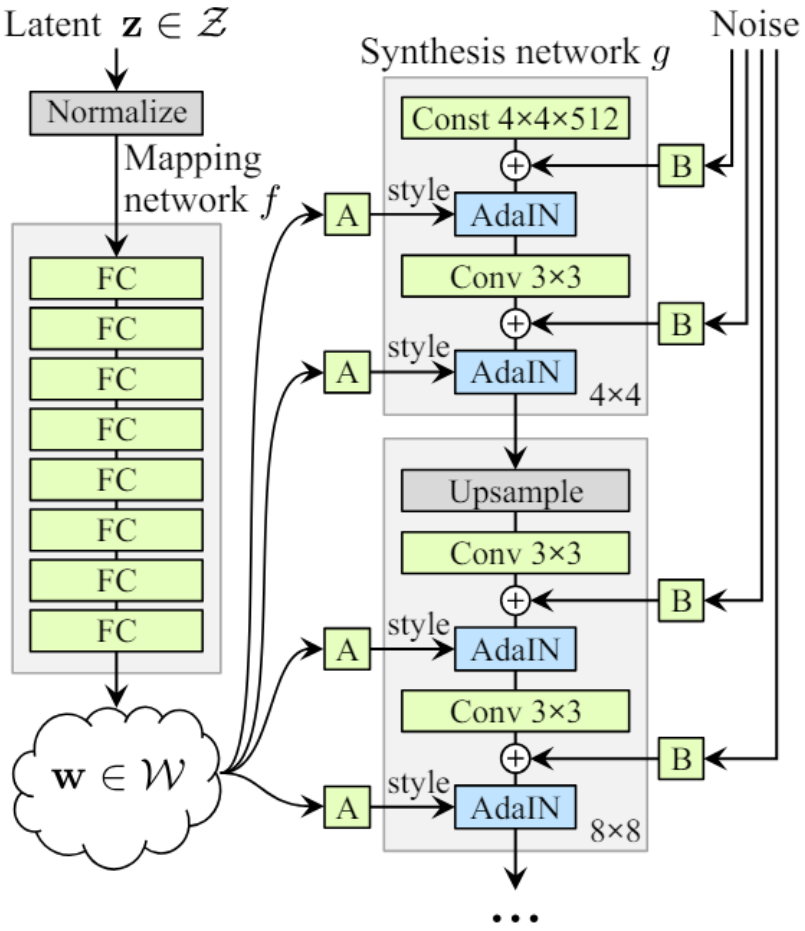

Figura 12 Generador Style-based Fuente: Karras, T. et al., 2019

Además de los bloques de mapeo de estilo, el generador también utiliza técnicas de interpolación y manipulación de estilo para permitir un control más preciso sobre las características de las imágenes generadas. Esto se logra mediante la combinación de múltiples vectores de ruido gaussiano no correlacionado y estilos latentes para producir imágenes con una variedad de estilos y características. Una técnica importante es la

regularización de la longitud del camino (Path Length), que penaliza la distancia en el espacio latente entre puntos vecinos, fomentando una distribución más suave y coherente de las características en el espacio latente.

3.2.8. StyleGAN3

StyleGAN3 (Karras et al., 2021), la última iteración de la serie StyleGAN, ha sido desarrollada con el objetivo de abordar las limitaciones y desafíos encontrados en versiones anteriores, en especial el problema de aliasing, al tiempo que impulsa aún más los límites de la generación de imágenes. Al aprovechar avances en técnicas de aprendizaje profundo y arquitecturas de redes neuronales, StyleGAN3 promete ofrecer imágenes aún más realistas y detalladas, con una eficiencia mejorada en términos de recursos computacionales y tiempo de entrenamiento.

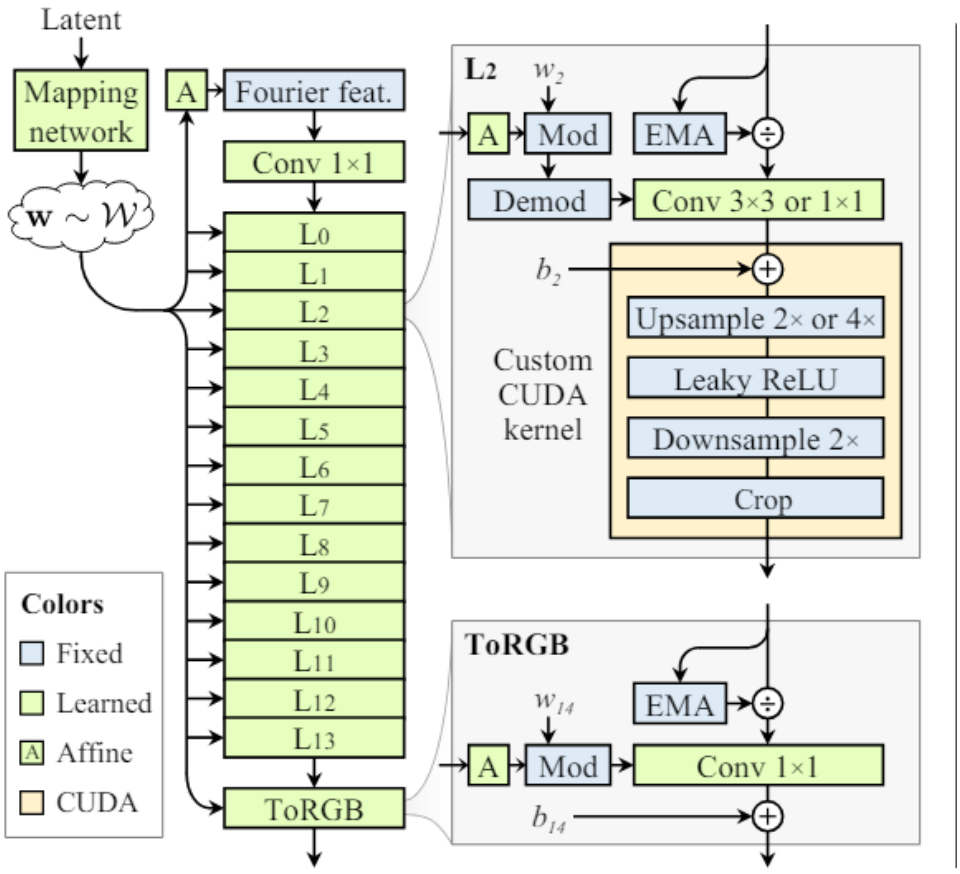

Figura 13 Arquitectura del generador del alias-free StyleGAN3. Fuente: Karras, T. et al., 2021

El aliasing es un fenómeno visual que ocurre cuando una imagen digitalizada o una señal analógica son muestreadas a una frecuencia insuficiente, lo que produce artefactos o distorsiones no deseadas en la imagen final. Es especialmente evidente al rotar una imagen,

donde los píxeles parecen estar "pegados" a lugares específicos y no rotan de manera natural. Para abordar este desafío, StyleGAN 3 implementó una serie de mejoras en el generador con el objetivo de eliminar el efecto de aliasing de las imágenes generadas. Esto se logró haciendo que cada capa de la red de síntesis emitiera una señal continua, permitiendo transformar los detalles de manera conjunta.

Algunas de las mejoras clave incorporadas en el generador de StyleGAN3 incluyen: la sustitución de métricas para la relación señal-ruido pico (PSNR) y una métrica similar EQ-R para rotaciones, la sustitución de la constante de entrada en StyleGAN 2 con características de Fourier para definir un mapa espacialmente infinito, la reducción de la profundidad de la red de mapeo, la eliminación de conexiones de salto de salida y la introducción de una nueva capa afín aprendida que produce parámetros de traslación y rotación global para las características de Fourier de entrada.

Además, StyleGAN3 introdujo una optimización específica compatible con CUDA, así como un truco de estabilización al inicio del entrenamiento para evitar que el discriminador se enfoque demasiado en frecuencias altas en las primeras etapas del proceso de entrenamiento. Estas mejoras combinadas tienen como objetivo mejorar la calidad, la estabilidad y la utilidad de StyleGAN 3 en la generación de imágenes realistas y la creación de videos y animaciones de alta calidad.

3.2.9. ProGAN

Las ProGAN (Progressive Generative Adversarial Networks, Karras et al., 2018) surgieron con la idea de resolver el problema que tienen los GAN para generar imágenes de alta resolución, ya que es más fácil para el discriminador detectar cuando una imagen es sintética, creada por el generador, cuanto mayor es la resolución. La arquitectura del ProGAN se caracteriza por hacer crecer el generador y el discriminador progresivamente, pasando de imágenes de baja resolución a una alta resolución, al ir añadiendo capas que introduzcan detalles según va avanzando el entrenamiento. De esta manera se consigue que la red se adapte a la complejidad de los datos, para mejorar la estabilidad en imágenes de alta resolución.

Además, un beneficio adicional de aplicar el crecimiento progresivo es la reducción del tiempo de entrenamiento dado que gran parte de las iteraciones se realiza con resoluciones

bajas, con lo que se obtienen los mismos resultados en cuanto a calidad mucho más rápido, dependiendo de la resolución final por supuesto.

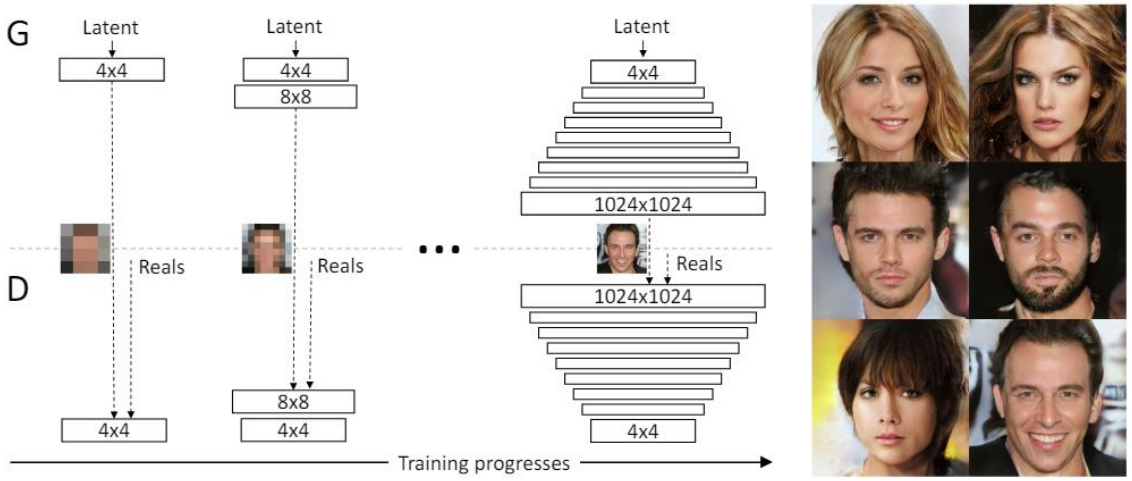

Figura 14 Arquitectura ProGAN Fuente: Karras, T. et al., 2017

3.2.10. FastGAN

El problema que se aborda utilizando el método FastGAN (Liu et al., 2021) es el de generar imágenes de alta resolución teniendo disponibles sólo un conjunto de datos limitado para entrenar, sin aumentar el esfuerzo computacional necesario para conseguirlas. En estas condiciones, con pocas imágenes, existe un alto riesgo de sobreajuste y colapso modal.

Para poder superar estos problemas en estas condiciones el generador tiene que entrenarse de una manera muy rápida, y el discriminador debe está entregando constantemente información útil que sirva al generador.

Se utiliza el módulo SLE (Skip-Layer channel-wise Excitation, Liu et al., 2021) que se encarga de revisar las respuestas proporcionadas por el canal en mapas de características de alta escala, como imágenes de alta resolución, aprovechando la información que le proporcionan activaciones de baja escala. De esta manera el flujo del gradiente es más robusto en todos los pesos y se consigue terminar antes el entrenamiento. Además, se consigue que aprenda una separación entre el estilo y el contenido, de igual manera que ocurría con el modelo StyleGAN.

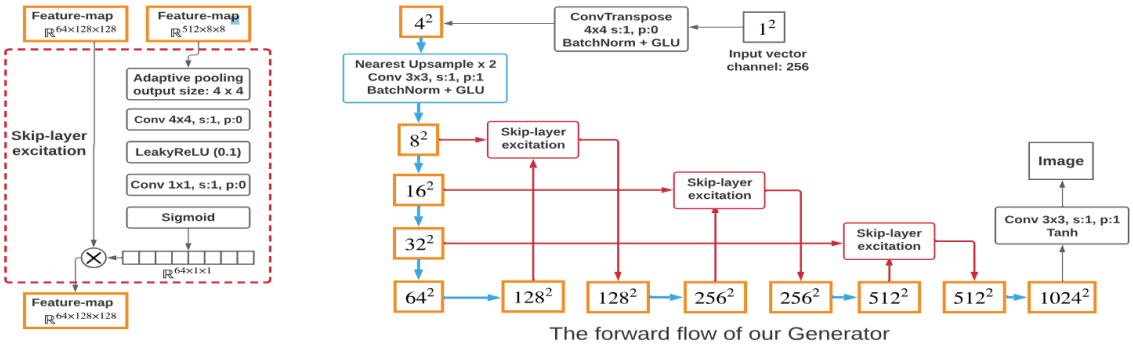

Figura 15 Estructura del módulo Skip-layer y del Generador de una FastGAN. Fuente: Liu, B. et al., 2020

Las cajas naranjas representan mapas de características (mostramos el tamaño espacial y omitimos el número de canales), la caja y las flechas azules representan la misma estructura de aumento de muestreo, la caja roja contiene el módulo SLE (Spatially Localized Enhancement, Mejora espacialmente localizada) como se ilustra a la izquierda.

Otra técnica que se utiliza en este método es la utilización de un discriminador auto supervisado, que se entrene como codificador de funciones de un descodificador adicional. De esta manera el discriminador no solo determinaría si la imagen es real o sintética, sino que además obtiene un mapa de características más descriptivo al cubrir más regiones de la imagen. Esto se utilizará para proporcionar una retroalimentación al generador mucho más completa, guiándolo de manera más efectiva hasta resultados de mayor calidad.

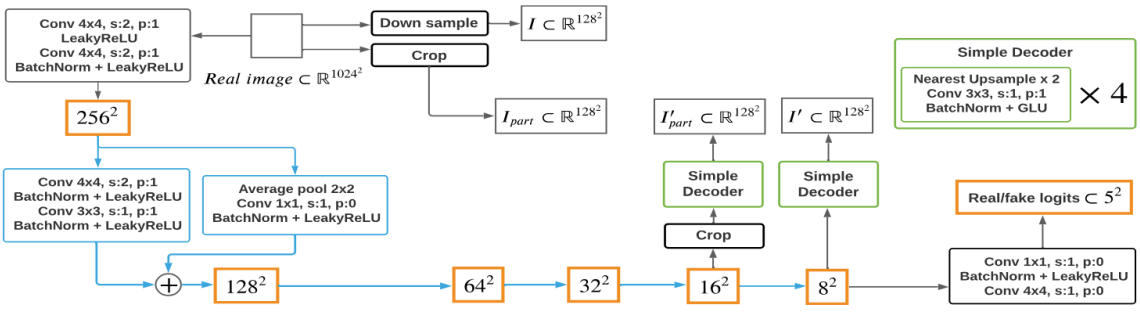

Figura 16 La estructura y el flujo hacia adelante del Discriminador. Fuente: Liu, B. et al., 2020

La caja azul y las flechas representan la misma estructura de muestreo descendente residual, las cajas verdes significan la misma estructura de decodificador.

3.3. Conclusiones del estado del arte

Se ha destacado la importancia de ampliar la investigación sobre la generación de imágenes, y especialmente sobre la generación de imágenes aéreas. A continuación, se han expuesto los principales métodos de generación de imágenes, junto con algunos de los métodos más populares que han demostrado buenos resultados en otros ámbitos. Queda claro que con los métodos GAN se logran resultados de mayor calidad que con los métodos VAE. Dentro de las redes generativas adversarias sería interesante comparar: FastGAN[2], ProGAN[3] y StyleGAN3[4]. Dado que no se entrenarán en un entorno con altos recursos computacionales, una opción sería FastGAN que se comporta de manera eficiente en este tipo de entornos, consiguiendo con poco tiempo resultados de calidad competente. Evaluar su comportamiento en este contexto y ponerlo a prueba será de gran interés. ProGAN sería el modelo con más antigüedad de los tres, por lo que servirá para ver el desarrollo de los modelos. Ha demostrado buenos resultados utilizando la técnica de refinamiento progresivo aumentando la resolución, un método que también merece ser comparado. Por otro lado, StyleGAN 3 es el método más reciente, que mejora sobre StyleGAN 2, y es considerado uno de los mejores métodos actuales. Evaluarlo junto con otros modelos más antiguos permitirá ver su capacidad y apreciar sus mejoras de rendimiento. Todos estos algoritmos han sido capaces de generar imágenes de otros dominios que no sean aéreas, como de rostros u objetos, de manera favorables. Por lo tanto, será interesante estudiar su comportamiento en la generación de imágenes aéreas. Son modelos que se encuentran disponibles para uso particular, gracias a que los creadores permiten su aplicación de manera gratuita, con su correspondiente citación. Por todos estos motivos, se ha decidido que estos algoritmos los que se comparen en este proyecto.

[2] FastGAN: https://github.com/odegeasslbc/FastGAN-pytorch
[3] ProGAN: https://github.com/facebookresearch/pytorch_GAN_zoo
[4] StyleGAN 3: https://github.com/NVlabs/stylegan3

4. Planteamiento de la comparativa

En este apartado se llevará a cabo el planteamiento de cómo se va a realizar la comparativa entre los modelos seleccionados para su estudio en la generación de imágenes sintéticas aéreas. Se determinará de qué manera se va a realizar, a través de que plataforma de cómputo se van a ejecutar los entrenamientos. Se seleccionará el dataset de entrenamiento, los parámetros con los que trabajarán los modelos y las diferentes métricas o métodos de evaluación que se utilizarán para evaluar los resultados.

4.1. PLATAFORMAS EN LA NUBE PARA EL ENTRENAMIENTO DE MODELOS

Para entrenar modelo de generación de imágenes se necesita una infraestructura computacional robusta y flexible, y existen una serie de plataformas en la nube que intentan satisfacer esta necesidad. Entre ellas se encuentran Microsoft Azure[5], Google Cloud Platform[6] (GCP) y Amazon Web Services[7] (AWS), que proporcionan entornos virtuales ideales para llevar a cabo estas actividades, ofreciendo una amplia variedad de recursos escalables, potentes y accesibles que permiten seleccionar las mejores características para tu entorno de entrenamiento. Nvidia[8] y IBM[9] también han desarrollado plataformas, más dedicadas a empresas y profesionales, pero que también son adecuadas para esta tarea. En este apartado se examinarán las características, ventajas y desventajas de estas plataformas líderes, además de comparar sus capacidades. En los siguientes artículos se puede encontrar información sobre estas plataformas y su comparación: DataWolke, 2023; Clarat, 2024; Manjaly, S., 2022; Martynek R., 2023.

Aunque si es cierto que finalmente no se ha utilizado ninguna de estas plataformas para la comparativa de este TFM, se han barajado todas estas opciones y se ha realizado una investigación de las diferentes opciones posibles para realizarlo a través de ellas. Por desgracia, aunque hubiera sido lo mejor dadas las posibilidades que ofrecen, debido a temas económicos no ha sido posible. Ninguna de las plataformas ofrece un entorno gratuito con

[5] Microsoft Azure: https://azure.microsoft.com/es-es
[6] GCP: https://cloud.google.com/?hl=es_419
[7] AWS: https://aws.amazon.com/es/
[8] Nvidia: https://www.nvidia.com/es-es/data-center/gpu-cloud-computing/
[9] IBM: https://www.ibm.com/es-es

las propiedades requeridas para entrenar modelos de generación de imágenes ya que no permiten disponer de GPU.

4.1.1. Microsoft Azure

Microsoft Azure[10] es una plataforma de servicios en la nube que ofrece un amplio rango de herramientas y servicios para el desarrollo de IA, como Azure Machine Learning (Azure ML), Azure Databricks y una gran variedad de tipos de máquina virtual (Virtual Machines, VM) optimizadas para cargas de trabajo intensivas.

Azure ML está diseñado para facilitar la creación, el entrenamiento y la implementación de modelos de ML. Ofrece capacidades de autoML, integración con Jupyter Notebooks y soporte para frameworks populares como TensorFlow, PyTorch y scikit-learn. La integración con Azure DevOps también permite un flujo de trabajo CI/CD eficiente para el despliegue continuo de modelos.

Ventajas	**Desventajas**
Integración profunda con herramientas de Microsoft, como Visual Studio y GitHub.	Puede resultar complejo de configurar y administrar para usuarios sin experiencia previa en la plataforma.
Fuerte soporte para flujos de trabajo de DevOps y CI/CD.	Costos relativamente altos en comparación con otras plataformas, especialmente para instancias de VM de alto rendimiento.
Servicios avanzados de gestión de experimentos y recursos.	

Tabla 1 Ventajas y desventajas de Azure Fuente: Choudhary, A et al., 2022

Una de las características distintivas de Azure ML es su capacidad para gestionar experimentos de manera eficiente. Los usuarios pueden ejecutar múltiples experimentos simultáneamente, comparar resultados y seleccionar los mejores modelos basándose en

[10] https://learn.microsoft.com/en-us/azure/?product=popular

métricas específicas. Además, Azure ML facilita la administración de recursos a través de clústers de computación que pueden escalar automáticamente según la demanda, optimizando así el uso de recursos y reduciendo costos.

Azure Databricks, una colaboración entre Microsoft y Databricks, proporciona un entorno de análisis y aprendizaje automático que integra Apache Spark[11]. Esta plataforma permite la preparación y el procesamiento de grandes volúmenes de datos de manera eficiente.

4.1.2. Google Cloud Platform (GCP)

Google Cloud Platform (GCP)[12] es otra opción conocida para el entrenamiento de modelos de IA. GCP ofrece un catálogo completo de servicios de IA, incluyendo Google AI Platform, Google BigQuery y TensorFlow.

Google AI Platform es el servicio principal de GCP para el desarrollo de modelos de IA. Facilita el entrenamiento, la validación y la implementación de modelos de aprendizaje automático y aprendizaje profundo. AI Platform soporta frameworks como TensorFlow, Keras, PyTorch y scikit-learn, permitiendo a los desarrolladores utilizar las herramientas que mejor se adapten a sus necesidades.

Ventajas	Desventajas
Acceso a TPUs, que pueden acelerar significativamente el entrenamiento de modelos de IA.	Puede ser costoso, especialmente al utilizar TPUs.
Excelente integración con TensorFlow y otras herramientas de Google.	La curva de aprendizaje puede ser empinada para usuarios nuevos en la plataforma.
Amplias capacidades de análisis y procesamiento de datos con BigQuery.	

Tabla 2 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022

[11] https://spark.apache.org/
[12] https://cloud.google.com/docs?hl=es-419

Además, GCP ofrece TPUs (Tensor Processing Units), hardware especializado desarrollado por Google para acelerar el entrenamiento de modelos de IA. Las TPUs pueden proporcionar mejoras de rendimiento significativas en diversas situaciones, en comparación con las GPUs tradicionales (My Grafic Card, 2024).

BigQuery es otra herramienta poderosa de GCP, que permite el análisis y procesamiento de grandes conjuntos de datos de manera rápida y eficiente. Para proyectos que requieren la manipulación de grandes volúmenes de datos, BigQuery puede ser extremadamente útil.

4.1.3. Amazon Web Services (AWS)

Amazon Web Services (AWS)[13] es una de las plataformas en la nube más completas y ampliamente utilizadas en la industria. AWS en la extensa gama de servicios que ofrece destacan Amazon SageMaker, EC2 (Elastic Compute Cloud) y S3 (Simple Storage Service).

Amazon SageMaker es el servicio estrella de AWS para el aprendizaje automático. Proporciona un entorno integrado que facilita cada paso del proceso de desarrollo de modelos de IA, desde la preparación de datos hasta el entrenamiento, la optimización y la implementación. SageMaker soporta una variedad de frameworks de aprendizaje automático y aprendizaje profundo, incluyendo TensorFlow, PyTorch, MXNet y otros.

SageMaker destaca por su capacidad para automatizar gran parte del flujo de trabajo de aprendizaje automático. Por ejemplo, SageMaker Autopilot puede automatizar el proceso de selección de modelos y ajuste de hiperparámetros, lo que facilita el trabajo a los desarrolladores. Además, ofrece herramientas avanzadas de monitorización y gestión de modelos, facilitando el seguimiento del rendimiento y la gestión de despliegues en producción.

EC2 proporciona instancias de VM con una amplia variedad de configuraciones, desde instancias de propósito general hasta instancias optimizadas para computación intensiva con GPUs de alto rendimiento, como las P3 y P4. Estas instancias son ideales para el entrenamiento de modelos de IA que requieren gran capacidad de procesamiento.

[13] https://docs.aws.amazon.com/

S3 es el servicio de almacenamiento de objetos de AWS y es altamente escalable y duradero siendo una opción excelente para almacenar grandes conjuntos de datos.

Ventajas	Desventajas
Amplia gama de servicios y herramientas para cada etapa del desarrollo de IA.	La estructura de precios puede ser compleja y difícil de prever.
Alta flexibilidad y escalabilidad, con opciones de instancias de VM optimizadas para diversas necesidades.	La gran cantidad de opciones y configuraciones puede ser abrumadora para los usuarios novatos.
Soluciones avanzadas de automatización y gestión de modelos con SageMaker.	

Tabla 3 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022

4.1.4. Nvidia

Nvidia[14] es un líder en el desarrollo de hardware y software para IA, proporcionando plataformas robustas específicamente diseñadas para tareas de aprendizaje profundo. Nvidia ofrece diversas soluciones en la nube a través de su Nvidia GPU Cloud (NGC) y la plataforma Nvidia DGX.

Nvidia GPU Cloud (NGC) es un catálogo de software que incluye contenedores optimizados para IA, aprendizaje profundo y análisis de datos. NGC proporciona acceso a los frameworks más populares, así como a herramientas de visualización y análisis de datos. Los contenedores están optimizados para ejecutarse en la infraestructura de Nvidia, incluyendo GPUs en la nube, lo que facilita el entrenamiento y despliegue de los modelos.

Nvidia DGX es una serie de sistemas de hardware diseñados específicamente para tareas de IA. Los sistemas DGX combinan potentes GPUs Nvidia con software optimizado para maximizar el rendimiento en tareas de aprendizaje profundo. La serie DGX incluye el DGX Station para entornos de oficina y el DGX-2, que es un sistema de servidor altamente escalable. Estos sistemas están diseñados para proporcionar el máximo rendimiento en

[14] https://www.nvidia.com/en-us/data-center/

entrenamiento de modelos de IA, haciendo uso eficiente de la potencia computacional de las GPUs de Nvidia.

Ventajas	Desventajas
Hardware altamente optimizado para tareas de IA, proporcionando un rendimiento superior.	Costos iniciales elevados para hardware DGX.
Contenedores de software preconfigurados y optimizados para aprendizaje profundo en NGC.	Puede requerir una inversión significativa en infraestructura para aprovechar todas las capacidades de Nvidia.
Soluciones escalables desde estaciones de trabajo hasta grandes centros de datos.	

Tabla 4 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022

4.1.5. IBM

IBM[15] ha sido un pionero en el desarrollo de soluciones de inteligencia artificial, proporcionando plataformas en la nube, como IBM Watson y IBM Cloud, que soportan el desarrollo y entrenamiento de modelos de IA.

IBM Watson es una suite de herramientas y servicios de IA que permite a los desarrolladores crear, entrenar y desplegar modelos de aprendizaje automático. Watson Studio es una plataforma integrada que proporciona un entorno colaborativo para científicos de datos y desarrolladores, facilitando el entrenamiento y la implementación de modelos. Watson Studio soporta una variedad de frameworks y herramientas, incluyendo TensorFlow, PyTorch y scikit-learn, al igual que sus competidores.

IBM Cloud ofrece una infraestructura escalable para el desarrollo y despliegue de aplicaciones de IA, incluyendo instancias de VM optimizadas para cargas de trabajo de IA, almacenamiento en la nube y herramientas de análisis de datos. La integración con Watson

[15] https://www.ibm.com/docs/en

permite a los usuarios aprovechar las capacidades avanzadas de IA de IBM en una infraestructura flexible y escalable.

Ventajas	Desventajas
Soluciones de IA avanzadas con Watson, proporcionando herramientas potentes para el desarrollo de modelos.	La curva de aprendizaje puede ser empinada para usuarios nuevos en la plataforma.
Infraestructura escalable con IBM Cloud, soportando una amplia gama de necesidades de computación.	Los costos pueden ser elevados, especialmente para servicios avanzados de IA.
Amplias capacidades de análisis y gestión de datos.	

Tabla 5 Ventajas y desventajas de GCP Fuente: Choudhary, A et al., 2022

4.1.6. Problema con las plataformas en la nube

El uso de una plataforma de entrenamiento en la nube no ha sido posible en este proyecto debido a limitaciones económicas. No disponemos de recursos para comprar ni acceder a ninguno de los servicios que ofrecen estas plataformas. Se ha considerado la posibilidad de utilizar las suscripciones gratuitas que ofrecen, las cuales permiten acceso limitado a algunos de sus servicios por un tiempo determinado. Sin embargo, estas suscripciones solo permiten acceso a una serie de recursos muy restringidos.

Para entrenar modelos de generación de imágenes, es necesario disponer de GPUs y una alta capacidad de memoria, ya que estos entrenamientos requieren un rendimiento significativo. Sin estos recursos, utilizando solo CPU, los tiempos de entrenamiento se prolongan considerablemente, haciendo que esta práctica sea poco eficiente. Al estudiar los servicios gratuitos que ofrecen estas plataformas, se constató que ninguna de ellas ofrece GPUs de forma gratuita, solo permitían acceso a CPU, por lo que no es viable entrenar estos modelos en dichas plataformas en este proyecto.

La experiencia al trabajar con estas plataformas no ha sido muy satisfactoria. En general, se caracterizan por ser poco accesibles e intuitivas, lo que complica el acceso a los servicios

buscados. En las versiones gratuitas, es difícil entender a qué partes se tiene acceso debido a las severas limitaciones. Además, cada plataforma tiene sus propias particularidades, lo que ha requerido un tiempo considerable para familiarizarse con cada una, con resultados generalmente negativos. Si se dispusiera de recursos económicos, estas plataformas podrían ser muy valiosas, ya que ofrecen una potencia computacional significativa por el tiempo que el usuario necesite. Sin embargo, en el caso de acceder solo a los recursos gratuitos, la oferta es extremadamente limitada, resultando insuficiente para realizar tareas que requieran un nivel de complejidad.

4.2. ENTORNO DEL ENTRENAMIENTO LOCAL

Debido no poder realizar el entrenamiento a través de una plataforma en la nube, se realizarán los entrenamientos de los modelos en el entorno controlado de un ordenador personal. A continuación, se detallan las características del ambiente de entrenamiento:

Especificaciones del Hardware:

- Modelo del Portátil: ASUS TUF Dash F15 FX517ZM_FX517ZM
- Procesador: Intel(R) Core (TM) i7-12650H de 12ª generación con 10 núcleos (6 de alto rendimiento y 4 de eficiencia), 16 hilos, frecuencia base de 2.30 GHz y turbo de hasta 4.70 GHz.
- Memoria RAM: 16 GB DDR5 a 4800 MHz.
- Tarjeta Gráfica: NVIDIA GeForce RTX 3060, con 6 GB de VRAM GDDR6.
- Almacenamiento: Unidad de estado sólido (SSD) NVMe de 453 GB.
- Sistema Operativo: Windows 11 Home, versión 23H2.

Software y Herramientas Utilizadas:

- Bibliotecas de Aprendizaje Automático: PyTorch 2.3.0, con la versión de CUDA 12.1
- Entorno de Desarrollo Integrado (IDE): PyCharm Community Edition 2021.2.

4.3. DATASET

Para realizar la comparación se han barajado varios posibles datasets. Uno de los más utilizados comúnmente en el aprendizaje profundo con imágenes aéreas es Map2map (Marra et al., 2018) que cuenta con 2000 imágenes satelitales, hechas mediante Google maps, con lo que cuentan con datos de carreteras. Es fácilmente accesible dado que está

incluido en TensorFlow 2.0. Se creó para ser utilizado con CycleGAN (Zhu et al., 2017), un generador dedicado a transformar imágenes entre dos estilos, por lo que para este tipo de empleos es ideal, pero su tamaño no es muy grande, tiene una falta de diversidad y una baja resolución, por lo que no es adecuado para otras tareas.

AID (Xia et al., 2017) es un dataset creado para los entrenamientos de clasificadores de imágenes, por lo que cuenta con pequeños conjuntos, de 300 imágenes de resolución 600x600 píxeles cada uno, de 30 categorías diferente como: parques, ríos, campos de béisbol, playas, etc. Por lo que cuenta con una variedad muy grande, aunque con una resolución media. Son imágenes extraídas en muchos países de todo el mundo a través de Google Earth en varias épocas y en diferentes condiciones.

El conjunto de imágenes de INRIA (Maggiori et al., 2017) sería un buen ejemplo de dataset. Representa 810 km^2 de terreno, principalmente zonas urbanas de varias localizaciones geográficas como zonas muy pobladas de San Francisco, o zonas más rurales de Austria, ya que su uso original era para la detección de edificios. Son imágenes de alta resolución, en formato GeoTIFF y de acceso abierto, lo que lo convierte en un buen conjunto de datos para generar imágenes sintéticas.

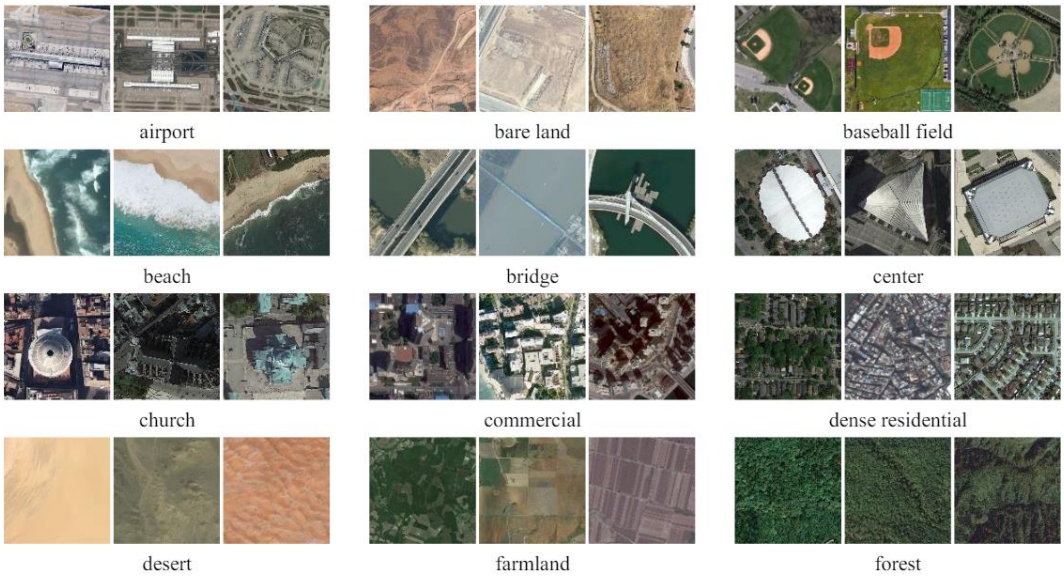

Figura 17 Ejemplos de las categorías en el conjunto de datos AID Fuente: Xia, G. et al., 2017

Al final el dataset por el que se ha optado ha sido el de Forest Aerial Images (Demir et al., 2018). Este dataset cuenta con 5108 imágenes, principalmente de bosques y terrenos montañosos de distintos tipos, además de contar con imágenes de ríos y edificaciones

rurales. Este dataset se utilizó en Land Cover Classification Track en DeepGlobe Challenge, un concurso dedicado avanzar en el análisis de imágenes satelitales a través de aprendizaje profundo en 2018. La resolución de las imágenes es de 256x256 pixeles, la cual es una baja resolución, pero suficiente para poder realizar la comparación y reducir el tiempo de entrenamiento. Se pueden ver más ejemplos en el Anexo A.

Figura 18 Ejemplos de imágenes del dataset de Forest Aerial Fuente: Demir, I. et al., 2018

El motivo por el que se ha elegido este conjunto de imágenes es porque cumple con todos los requisitos de una manera más optima que el resto. Dispone de un número de imágenes adecuado, sin ser demasiado pequeño, como le ocurre al dataset Map2map que solo cuenta con 2000 imágenes. La resolución de las imágenes es la idónea para poder trabajar en el entorno de entrenamiento seleccionado sin aumentar excesivamente la complejidad del proceso. No es una resolución muy alta, pero si se pueden diferenciar los diferentes elementos que forman parte la composición de las imágenes. Además, cuenta con una diversidad de imágenes en las que se pueden apreciar diferentes elementos, como rio, edificios, bosques, campos de cultivo o carreteras, lo que hace que se un dataset con diversidad y variedad. En el caso del conjunto de imágenes de AID, al tratarse de un dataset pensado para modelos de clasificación de imágenes, cuenta con demasiadas imágenes de categorías diferentes, lo que llevaría a un problema de entrenamiento con datos no heterogéneos, sin correlación entre sí, llegando a resultados no válidos.

4.4. MÉTRICAS

Por desgracia las redes generativas adversarias no disponen de una función objetivo, lo que significa que comparar el rendimiento de diferentes modelos no es tarea fácil. No existe una

forma generalizada para evaluar GAN, por lo que surgen problemas a la hora de investigar, como cuando: se debe parar de ejecutar el entrenamiento, se tienen que elegir las imágenes de muestra de la GAN, se compara la estructura de los modelos, o se comparan las diferentes configuraciones de los GAN.

Este sigue siendo un problema abierto de los modelos GAN, pero si se han intentado establecer diferentes medidas. Aunque nunca se ha llegado a un consenso sobre cuál de ellas es la que mejor capta las limitaciones y las ventajas de los modelos y debería ser la que se utilizara para determinar un ranking justo de los distintos modelos. Por lo tanto, en muchas ocasiones se evalúa su calidad en función al objetivo final con el que se hayan creado. (Brownlee J., 2019; Borji, 2019; Salimans et al., 2016)

4.4.1. Evaluación cualitativa

Debido a esto, muchas veces se recurre a la evaluación manual de las imágenes sintéticas creadas, lo que consiste en someter a personas a exámenes visuales de diferentes muestras y que sean estos los que evalúen la calidad de las imágenes. Esta prueba se puede realizar por los propios investigadores o por más persona. Serían exámenes en los que se presentan imágenes creadas sintéticamente e imágenes reales del mismo dominio, y los humanos son los que determinan la calidad y diversidad de estas. Como no se sabe cuándo se debe detener el entrenamiento, se suelen generar gran cantidad de imágenes en diferentes momentos para poder evaluar este aspecto también. Esta es una de las pruebas más intuitivas y comunes para evaluar las GAN.

Este es el método más sencillo de evaluación, pero presenta ciertos problemas:

- Subjetividad: incluye sesgos por parte de los observadores, sobre la configuración y el objetivo del proyecto.
- Es necesario conocer previamente lo que se entiende como realista y lo que no según el objetivo a tratar.
- Esta limitado por el número de imágenes que una persona puede clasificar en un determinado tiempo.
- El rendimiento de los jueces no es constante, es decir, pueden ir aprendiendo pistas a como sobre como detectar imágenes generadas.

Sobre todo, el problema de la subjetividad puede provocar que se haga una selección selectiva de las imágenes, generando sesgos y provocando que no sea un método ideal.

Los métodos de evolución que no utilizan métricas numéricas, sino que utilizan la subjetividad de las personas son métodos cualitativos. Los más utilizados serían:

- Evaluación y juicio de preferencia: Los jueces comparan y clasifican imágenes reales e imágenes generadas por la GAN.
- Categorización rápida de escenas: es igual que el anterior, salvo porque ahora las imágenes solo se muestran una fracción muy corta de tiempo.
- Vecinos más cercanos: Se muestra una imagen real y se tiene que seleccionar entre unas imágenes falsas las más similares a la real para compararla. Ayuda a determinar el realismo de imagen que se puede alcanzar.

Este tipo de evaluaciones suele requerir mucho trabajo, pero se puede deducir el costo haciéndolo a través de interfaces web, o facilitando la tarea mediante plataformas como Amazon Mechanical Turk[16], que ayuda a empresas a coordinar tareas que los ordenadores no son capaces de realizar de manera eficiente.

En este proyecto no se utilizará ningún método cualitativo para comparar los distintos generadores, salvo el criterio subjetivo para determinar en qué punto han sido lo suficientemente entrenadas las GAN y poder detenerlo de manera que se puedan completar los objetivos propuestos en el trabajo. No se realizará ninguna evaluación a través de estos métodos porque conllevaría un esfuerzo demasiado elevado, que no sería necesario, dado que utilizando diferentes métodos cuantitativos es realizar la comparación de manera más objetiva y sencilla, y con la validez necesaria para cumplir los objetivos establecidos. Aunque sí que se hará un comentario subjetivo en el análisis final para ayudar a las métricas cuantitativas y así añadir más valor y poder completar los detalles que estas no son capaces de cubrir.

4.4.2. Métricas cuantitativas

Las métricas cuantitativas hacen referencia a todas las que son puntuables a través de números para determinar la calidad de las imágenes generadas. Como se mencionó

[16] Amazon Mechanical Turk: https://www.mturk.com/

anteriormente, no existe un consenso general sobre cuales son las métricas ideales para evaluar este tipo de algoritmos, pero si que existen algunas cuyo uso está más generalizado y que han demostrado representar la calidad de las imágenes de manera muy confiable. Entre las más usadas están Frechet Inception Distance (FID), Inception Score (IS), Kernel Inception Distance (KID) y Geometric Score.

FID (Heusel, M. et al., 2017) es la métrica más popular y ampliamente aceptada por la comunidad, permite evaluar tanto la calidad como la diversidad, permitiendo evaluar de manera robusta la similitud entre la distribución de características de las imágenes generadas comparándolas con las reales, siendo poco sensible a la sobreestimación de la calidad. IS (Salimans et al. 2016) permite cuantificar como de distinta es cada imagen generada y como de variadas son las imágenes en términos de clases generadas, es decir, la confianza y la diversidad del conjunto de imágenes sintéticas de cada modelo. Es una métrica sencilla y rápida de usar, y muy aplicada en modelos que generan imágenes con variedad de elementos y escenarios. Con KID (Bińkowski et al., 2018) se cuantifica la discrepancia entre las distribuciones de características de las imágenes generadas y las reales usando el método kernel. De esta manera, no se asume ninguna distribución paramétrica que condicione las características, lo que genera un resultado más robusto en situaciones donde las distribuciones no son gaussianas, logrando resultados similares a FID, pero con menos sesgo. Geometric Score (Khrulkov y Oseledets, 2018) está especialmente diseñada para medir la diversidad de las imágenes generadas basándose en su distribución geométrica en el espacio de características, completando a otras métricas que se centran en la calidad. Proporciona una medida de cómo la diversidad de las imágenes generadas se distribuye en el espacio de características. Por todos estos motivos se han elegido estas métricas que se estudiarán más exhaustivamente a continuación:

4.4.2.1. Frechet Inception Distance (FID)

El FID (Heusel, M. et al., 2017) permite determinar la distancia entre las distribuciones de imágenes reales e imágenes generadas. Para conseguirlo, cuantifica la similitud entre las imágenes generadas y reales, a través de una Inception Network pre-entrenada (Simonyan, K. y Zisserman, A.,2014), extrae vectores de características y luego calcula la distancia de Férchet entre los dos conjuntos de vectores.

La ecuación que se utiliza para calcularlo sería la siguiente:

$$FID(r,g) = \|\mu_r - \mu_g\|_2^2 + T_r\left(\sum r + \sum g - 2(\sum r \sum g)^{\frac{1}{2}}\right)$$

Figura 19 Ecuación para calcular la distancia de Férchet

Donde μr es la media de las características de las imágenes reales y ∑r son las matrices de covarianza de las imágenes reales, mientras que μg y ∑g las de imágenes generadas respectivamente.

FID es una de las métricas más utilizadas a hablar de GAN dado que se ha demostrado que bastante consistente con el juicio humano (Heusel et al., 2017), además, es capaz de detectar la caída de modos, es decir, que detecta cuando un modelo está generando solo una imagen por clase, algo que por ejemplo Inception Score (Salimans et al. 2016) no puede conseguir. Es una buena herramienta de evaluación en términos de robustez, eficiencia computacional y discriminabilidad. Como desventaja, solo tiene en cuenta los dos primeros momentos de la distribución, ya que asume que las características tienen distribución gaussiana, algo que no siempre se cumple. Conseguir un valor bajo de FID significa que la distribución generada es similar a la distribución real.

4.4.2.2. Inception Score (IS)

Fue propuesto por Salimans et al. 2016 y utiliza una red neuronal pre-entrenada llamada Inception net (Simonyan, K. y Zisserman, A.,2014), al igual que FID, para extraer las características de altamente calificable y diversidad de las imágenes generadas respecto a las imágenes reales. Lo hace midiendo la divergencia KL promedio entre la distribución condicional de etiquetas p(y|x) de las muestras y la distribución marginal p(y) conseguida de entre todas las muestras, favoreciendo la baja entropía de p(y|x) pero una gran entropía de p(y). Como se expresa en la siguiente formula:

$$IS = \exp(E_x[KL(p(y|x)\|p(y))]) = exp(H(y) - E_x[H\langle y|x\rangle])$$

Figura 20 Ecuación para el cálculo de IS

donde p(y|x) es la distribución condicional de etiquetas para la imagen “x” estimada usando un modelo Inception pre-entrenado, H(x) representa la entropía de la variable “x”, KL es la divergencia de Kullback-Leibler y p(y) es la distribución marginal:

$$p(y) \approx {}^{1}/_{N} \sum_{n=1}^{N} p\langle y|x_n = G(z_n)\rangle$$

Figura 21 Ecuación de la distribución marginal

Está métrica es una de las más utilizadas y adaptadas en la evaluación de modelos GAN ya que permite establecer una relación de manera razonable entre la diversidad y la calidad de las imágenes sintéticas. Sin embargo, este método tiene alguna limitación:

- Favorece una "GAN de memoria", es decir, que alacena todas las muestras de entrenamiento, siendo incapaz de detectar sobreajuste, pudiendo llegar a ser engañado. Como no utiliza un conjunto de validación de reserva esta situación es más probables. (Yang et al., 2017)
- Puede ocurrir que no detecte si el modelo se ha atascado en un modo malo, es decir, ha llegado a un punto en el que es indiferente al colapso de modos.
- La red neuronal que utiliza IS fue entrenada con ImageNet, que cuenta con muchas clases de objetos. Esto puede hacer que favorezca más los modelos que generan buenos objetos y penalice los que se dedican a generar imágenes realistas.
- Es una medida asimétrica.
- La resolución de la imagen le afecta en el resultado final

Cuando se consiguen valores altos de IS indica que el modelo genera imágenes de alta calidad y con gran diversidad. En el análisis realizado por Zhou et al., 2017 se estudia de manera más intensa los aspectos de esta métrica.

4.4.2.3. Kernel Inception Distance (KID)

La Kernel Inception Distance (KID)[17] es una métrica basada en el cálculo de la disimilitud entre las distribuciones de las características extraídas del conjunto de imágenes reales y del de imágenes generadas (Bińkowski et al., 2018). Al igual que la Inception Score (IS) y la Fréchet Inception Distance (FID), la KID utiliza la red pre-entrenada Inception para extraer características significativas de las imágenes. Sin embargo, la KID se basa en el kernel trick para medir la distancia entre estas distribuciones de características. Esta métrica se basa en una versión del Test de Máxima Discrepancia Media (Maximum Mean Discrepancy, MMD) (Fortet et al., 1956), que es una medida de la diferencia entre dos distribuciones de probabilidad. La fórmula que aplica KID es la de MMD genérica aplicada a las imágenes:

[17] FID, IS y KID: https://github.com/toshas/torch-fidelity

$$KID^2 = \frac{1}{m(m-1)}\sum_{i\neq j} k(x_i, x_j) + \frac{1}{n(n-1)}\sum_{i\neq j} k(y_i, y_j) - \frac{2}{mn}\sum_{i,j} k(x_i, y_j)$$

Figura 22 Ecuación de máxima discrepancia media aplicada a imágenes

Donde x_i y x_j son las características extraídas de las imágenes reales, y_i y y_j son las características de las imágenes generadas, *k* es la función del kernel, normalmente de grado 3, *m* es el número de imágenes reales y *n* el de imágenes generadas.

A diferencia de la FID, la KID es una métrica insesgada, por lo que su valor no está influenciado por la cantidad de muestras, proporcionando una estimación precisa de la calidad de las imágenes generadas. Aunque al igual que las otras métricas, depende de las características extraídas de una red pre-entrenada en un conjunto de datos específico. Esto puede limitar su aplicabilidad a ciertos tipos de imágenes que difieren significativamente de las imágenes en ImageNet. Valores bajos de KID indican una mayor calidad en las imágenes generadas.

4.4.2.4. Geometric score

Creado por Khrulkov y Oseledets, 2018 se basa en comparar las propiedades geométricas de conjunto de datos subyacentes entre los datos reales y los datos generados[18]. Para poder explicar esto se requerirían muchos detalles técnicos por lo que solo se expondrá de manera intuitiva.

La idea principal es crear una representación gráfica, geométrica, de los datos usando la información de proximidad entre muestras, como puede ser, por ejemplo, las distancias entre pares de puntos. Para lograrlo se utiliza el concepto de complejo simplicial. Para entenderlo habría que imaginar que hay un conjunto de puntos, que serían los datos, y que esos puntos se conectan formando triángulos, tetraedros y otras figuras de dimensiones mayores, después, se cambiaría el umbral ϵ, que decide cuando conectar los puntos, por lo que cuanto más aumente, se generaran más conexiones y más figuras.

Para cada valor de ϵ se analizan las propiedades topológicas del complejo simplicial, en especial las homologías. Estas representan agujeros en diferentes dimensiones, por ejemplo, en 1D es un lazo y en 2D es una cavidad. A medida que ϵ va variando, se mide el tiempo que

[18] Geometric Score: https://github.com/KhrulkovV/geometry-score

se mantienen estos agujeros y con estos se construye un código de barras, una firma, que permita visualizar su duración.

De esta manera, se puede calcular el tiempo en el que están presentes estos agujeros, y se puede calcular el promedio sobre el conjunto de los datos, dando lugar a los tiempos de vida relativos medios (MRLT, Mean Relative Living Times). Entonces, se calcula el valor de MRLT para el conjunto de datos reales y el conjunto de datos generados, y por último con esos valores se calcula distancia L2 entre ellos. Esta distancia indica la similitud topológica entre los dos conjuntos. Cuanto más pequeño sea el resultado final de L2, indica mayor similitud topológica habrá entre las imágenes reales y las falsas.

4.5. PARÁMETROS DEL ENTRENAMIENTO

En este apartado se verá la configuración y los hiperparámetros seleccionados para realizar el entrenamiento de los modelos GAN. No se han modificado las opciones que venían predeterminados salvo para:

- Establecer el número de iteraciones.
- Adaptar los modelos al dataset de imágenes de 256x256.
- Ajustar el tamaño del lote, el batch, para optimizar el uso de la memoria de la GPU disponible.

Para determinar los resultados finales de los parámetros que se han cambiado, se han realizado diferentes pruebas que han consistido principalmente en realizar pequeños entrenamientos, de pocas iteraciones, cambiando las diferentes variables y observando que cambios ocurrían en los tiempos de entrenamiento y en las imágenes generadas.

Para determinar el tamaño óptimo de las imágenes con las que se iba trabajar se ha ido jugando con el tamaño del batch, el número de imágenes que procesa a la vez, valorando el tiempo de entrenamiento que emplea en cada caso. También se valorará como afecta a la cantidad de memoria de la GPU dedicada en cada caso. En este caso se cuenta con 6 GB de memoria dedicada, por lo tanto, la intención es aprovechar al máximo este espacio para procesar el mayor número de imágenes simultáneamente. Si el número el batch es demasiado alto como para saturar la memoria, el tiempo de entrenamiento aumentara considerablemente, mientras que, si se no es suficiente, no se utilizaran todos los recursos disponibles de manera eficiente.

Se realizarán pruebas con resoluciones de 128x128, 256x256, 512x512 y 1024x1024 píxeles, porque son las resoluciones que admiten los tres modelos. Los batch con los que se iterará serán de 6, 8 y 12 imágenes por el mismo motivo.

Batch size = 6

Resolución	**128x128**		**256x256**		**512x512**		**1024x1024**	
Tiempo/GPU	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**
FastGAN	1.29	3.2	1.45	4.4	1.25	5.8	8.29	5.8
StyleGAN3	63.38	1.4	85.04	2.1	139.85	3.8	312.58	5.8

Tabla 6 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 6 para las distintas resoluciones de imagen

Batch size = 8

Resolución	**128x128**		**256x256**		**512x512**		**1024x1024**	
Tiempo/GPU	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**
FastGAN	0.68	4.9	1.09	5.8	6.05	5.8	17.38	5.8
StyleGAN3	110.24	3.6	155.82	5.6	329.8	5.8	842.32	5.8

Tabla 7 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 8 para las distintas resoluciones de imagen

Batch size = 12

Resolución	**128x128**		**256x256**		**512x512**		**1024x1024**	
Tiempo/GPU	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**	**s/it**	**GB**
FastGAN	2.86	5.8	3.66	5.8	18.84	5.8	**	**
StyleGAN3	162.24	4	308.92	5.8	1533.84	5.8	**	**

Tabla 8 Tiempo de entrenamiento y uso de memoria de GPU con un batch de 12 para las distintas resoluciones de imagen

En los resultados obtenidos en las pruebas, se puede observar que al utilizar un batch de 6, una cantidad significativa de memoria de la GPU permanece vacía, Esto implica que los recursos computacionales no se están aprovechando de manera óptima durante el proceso de entrenamiento, lo que reduce la eficiencia del entrenamiento para todas las resoluciones. En el caso de utilizar un batch de 8, se puede apreciar como la memoria de la GPU alcanza su punto de uso máximo con las imágenes de resolución 256x256 y 512x512 píxeles. En este punto se emplea toda la memoria disponible, de la manera más eficiente, sin saturarla, logrando así el mayor rendimiento en términos de tiempo de entrenamiento. Al aumentar el batch a 12, salvo para la resolución de 128x128 píxeles, se excede la capacidad computacional disponible. Particularmente, con resolución de 1024x1024 se produce un error de memoria insuficiente. Por lo tanto, el valor del batch con el que mejor rendimiento se trabajaría es de 8 para imágenes de resolución 256x256 píxeles.

Los resultados obtenidos al realizar las pruebas con ProGAN no se pueden poner en el mismo formato que sus competidores debido a su particular metodología de entrenamiento. Este modelo incrementa progresivamente la resolución de las imágenes durante el entrenamiento. Al comenzar con resoluciones bajas, el uso de recursos computacionales es mínimo y las iteraciones son rápidas. Sin embargo, a medida que la resolución aumenta, también lo hacen los recursos necesarios y el tiempo de entrenamiento por iteración. Por lo tanto, para determinar el batch que se utilizará en este modelo, también se han realizado pruebas realizando entrenamiento con pocas iteraciones para determinar su valor. Dado que en los otros modelos el resultado final escogido para el batch ha sido de 8, y observando el

comportamiento de ProGAN con este mismo valor, se ha decido unificar el valor, ya que con el se logran valores razonables del tiempo de entrenamiento y se aprovechan eficientemente los recursos computacionales.

Como se mencionó anteriormente, el único aspecto en el que se aplicará un criterio cualitativo será para determinar el momento de detener los entrenamientos, es decir, el número de iteraciones que realizará cada modelo. Este criterio se basará en la mejora observada en las imágenes generadas durante las diferentes fases del entrenamiento y se prolongará hasta que no se detecte un aumento considerable en la calidad que justifique continuar con el entrenamiento del modelo.

Las variables que no requieran una determinación obligatoria se mantendrán con sus valores predeterminados. Esto se debe a que no solo podrían afectar la eficiencia en el tiempo de entrenamiento, sino también la calidad de los resultados. Por lo tanto, se ha decidido utilizar los valores establecidos por los creadores de los modelos, ya que estos permiten generar resultados con una calidad estándar que representa las capacidades inherentes de cada modelo para llevar a cabo esta función.

En general, las pruebas con los modelos han sido sencillas de realizar una vez conocido su funcionamiento. Es cierto que han surgido problemas al instalar todas las librerías y recursos que utilizan cada uno de los algoritmos. Se han encontrado numerosos inconvenientes relacionados con las diferentes versiones necesarias y su compatibilidad. Sin embargo, una vez que se lograron establecer todas las dependencias, el funcionamiento de los modelos ha sido satisfactorio y fácil de manejar.

4.5.1. FastGAN

Los únicos parámetros obligatorios que hay que indicar son la ruta al dataset, la dirección donde guardar los resultados, pero además se añadió: el nombre del entrenamiento, el número de iteraciones, la resolución de las imágenes del dataset y el número de GPUs a utilizar.

Se indicarán, en esta tabla y en el resto, con un * los parámetros que se han modificado y se dejarán en negro los predeterminados durante el entrenamiento.

Parámetro	Valor
Número de iteraciones (Iter)	100000*
Numero de GPUs (cuda)	1*
Resolución de las imágenes (Img_size)	256*
Tamaño del lote (batch_size)	8
Trabajadores para cargar datos (workers)	2
Frecuencia de guardado (save Interval)	100
Número de filtros en la 1º capa de D (ndf)	64
Número de filtros en la 1º capa de G (ngf)	64
Dimensión vector latente (nz)	256
Tasa de aprendizaje (nlr)	0.0002

Tabla 9 Valores de los parámetros del entrenamiento de FastGAN

4.5.2. ProGAN

Al entrenar ProGAN hay que indicar obligatoriamente la ruta donde guardar los resultados, el nombre del entrenamiento y la dirección a un archivo json, donde está indicada la ruta del dataset y la configuración de los hiperparámetros que se deseen modificar, si es que se desea cambiar alguno. Dentro de esta configuración, solo se modificó el tamaño de batch. Sobré el número de iteraciones, dado que ProGAN va aumentado la resolución con la que se entrena, se hizo hasta que alcanzo la misma resolución que las imágenes del dataset.

Parámetro	Valor
Tamaño del lote (miniBatchSize)	8*
Frecuencia de guardado (sabe_iter)	16000
Dimensión vector latente (dimLatentVector)	512
Inicios sesgos desde capa 0 (initBiasToZero)	True
Normalización por canal (perChannelNormalization)	True
Modo de pérdida (lossMode)	WGANGP
Penalización de WGANGP (lambdaGP)	10

Coeficiente de fuga (Leakyness)	0.2
Valor añadido a función de pérdida de D (epsilonD)	0.001
Desviación estándar (miniBatchStdDev)	True
Tasa de aprendizaje base (baseLearningRate)	0.001
Canales de salida (dimOutput)	3
Peso pérdida de G (weightConditionG)	0.0
Peso pérdida de D (weightConditionD)	0.0
Uso de penalización de divergencia GDPP	False
Número de iteraciones en cada escala de resolución (maxIterAtScale)	48000, 96000, 96000, 96000, 96000, 96000, 96000
Modo de transición de Alpha entre escalas (alphaJumpMode)	Linear
Número de saltos de Alpha entre escalas (alphaNJumps)	0, 600, 600, 600, 600, 600, 600,
Tamaño de salto de Alpha (alphaSizeJumps)	0, 32, 32, 32, 32, 32, 32
profundidad de las escalas (depthScales)	512, 512, 512, 512, 256, 128

Tabla 10 Valores de los parámetros del entrenamiento de ProGAN

4.5.3. StyleGAN3

Para entrenar StyleGAN3 se solicitan más parámetros obligatorios como la ruta donde guardar el entrenamiento, la ruta del dataset, el número de GPUs, el tamaño del batch, el número de iteraciones y la configuración de Stylegan3 que se quiere utilizar. También se cambió el snap, que cambia la frecuencia con la que se guardan los pesos del modelo.

Parámetro	**Valor**
Configuración de Stylegan3 (cfg)	Stylegan3-r*
Número de gpus (gpus)	1*
Tamaño del lote (batch)	8*
Factor de regulación en la pérdida de D (gamma)	2*

Número de iteraciones (kimg)	4000*
Máximo de canales por capa (cmax)	256*
Base para calcular el número de canales de características (cbase)	8192*
Frecuencia de guardado (snap)	20*
Modo de aumento (aug)	ada
Trabajadores para cargar datos (workers)	3
Tasa de aprendizaje de G (glr)	0
Tasa de aprendizaje de D (dlr)	0.002

Tabla 11 Valores de los parámetros del entrenamiento de StyleGAN3

4.6. CRITERIOS DE ÉXITO DE LA COMPARATIVA

Para poder afirmar que la comparativa ha sido exitosa, se deberán cumplir los siguientes criterios:

- Entrenamiento consistente: Todos los modelos se habrán entrenado de manera correcta en el mismo entorno de procesamiento, bajo las mismas condiciones, sin interferencia de variables externas que puedan comprometer la veracidad de los resultados.
- Aplicación eficiente de las métricas: Las métricas deben aplicarse a las imágenes generadas de manera efectiva, y los resultados obtenidos deberán estar dentro de los parámetros previstos, para afirmar que no ha habido problemas al aplicarlas que hayan provocada resultados erróneos
- Generación de imágenes claras: Los modelos habrán conseguido generar imágenes aéreas en las que se puedan diferenciar elementos concretos, dentro de las capacidades de cada modelo. Esto confirmará que la elección de estos modelos para su comparativa ha sido la acertada al utilizar modelos capaces de generar imágenes, independientemente de la calidad final.
- Evaluación comparativa: Los resultados obtenidos, tanto con las métricas, como con las imágenes generadas y el tiempo de entrenamiento, deberán proporcionar una idea clara del nivel de calidad de las imágenes generadas, que permita valorar los modelos individualmente, facilitando el poder compararlas entre sí.

5. DESARROLLO DE LA COMPARATIVA

Se realiza el entrenamiento de los modelos tal y como se ha especificado, ejecutando los diferentes modelos, y se realizan los cálculos de las métricas. Es este apartado se recogerán los resultados obtenidos para su posterior estudio y análisis.

5.1.IMÁGENES GENERADAS

Para realizar la evaluación de los generadores de imágenes, poder aplicar las métricas y realizar la comparativa, se han generado 5 conjuntos de imágenes de 5000 imágenes sintéticas cada uno con cada uno de los generadores. Aquí se pueden observar algunos ejemplos:

Figura 23 Ejemplo imágenes generadas con FastGAN

Figura 24 Ejemplo imágenes generadas con ProGAN

Figura 25 Ejemplo imágenes generadas con StyleGAN3

5.2. APLICACIÓN DE LAS MÉTRICAS

Se ha realizado el cálculo de las métricas sobre los 5 conjuntos de 5000 imágenes generadas por cada uno de los modelos a comparar en este proyecto y se ha realizado la media de cada uno de ellos. Los resultados obtenidos has sido:

Modelo	FID	IS	KID	L2
FastGAN	26.159	5.119 ± 0.148	0.006 ± 0.001	0.076
ProGAN	129.431	2.717 ± 0.058	0.099 ± 0.002	0.064
StyleGAN3	25.146	5.511 ± 0.180	0.005± 0.001	0.043

Tabla 12 Resultados de las métricas

El valor de FID, KID y L2 será más positivo cuanto menor sea, mientras que el valor de IS indicará mejores resultados cuanto mayor sea.

Al calcular Geometric Score también se han generado los gráficos de barras que representan los MRLT de las imágenes reales y generadas.

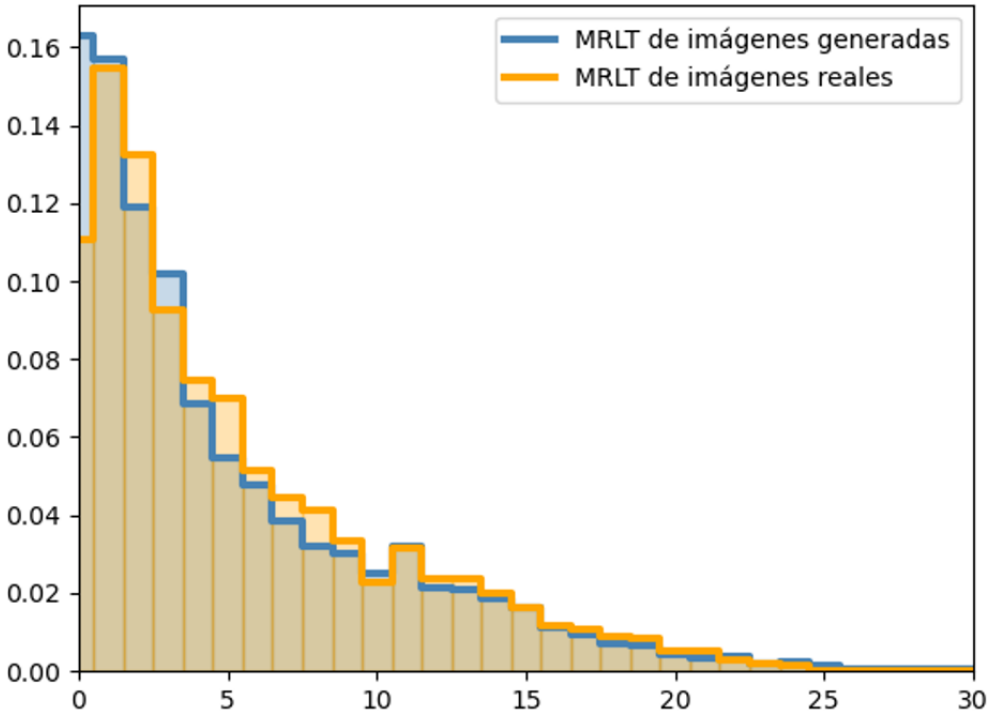

Figura 26 Gráfico con valores de MRLT de las imágenes reales y generadas de ProGAN

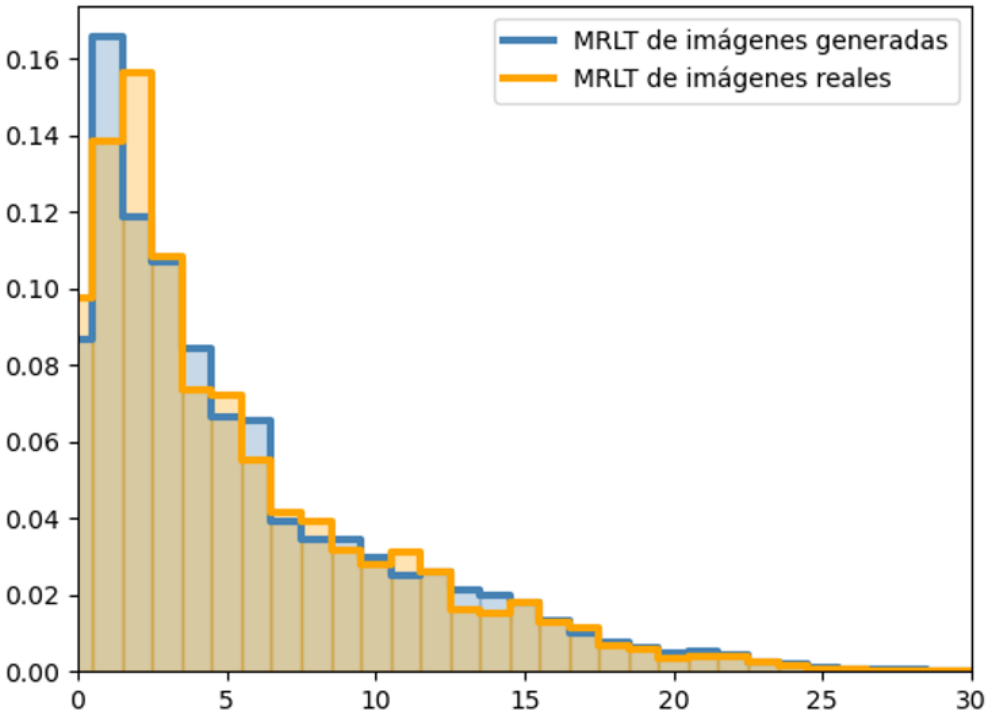

Figura 27 Gráfico con valores de MRLT de las imágenes reales y generadas de StyleGAN

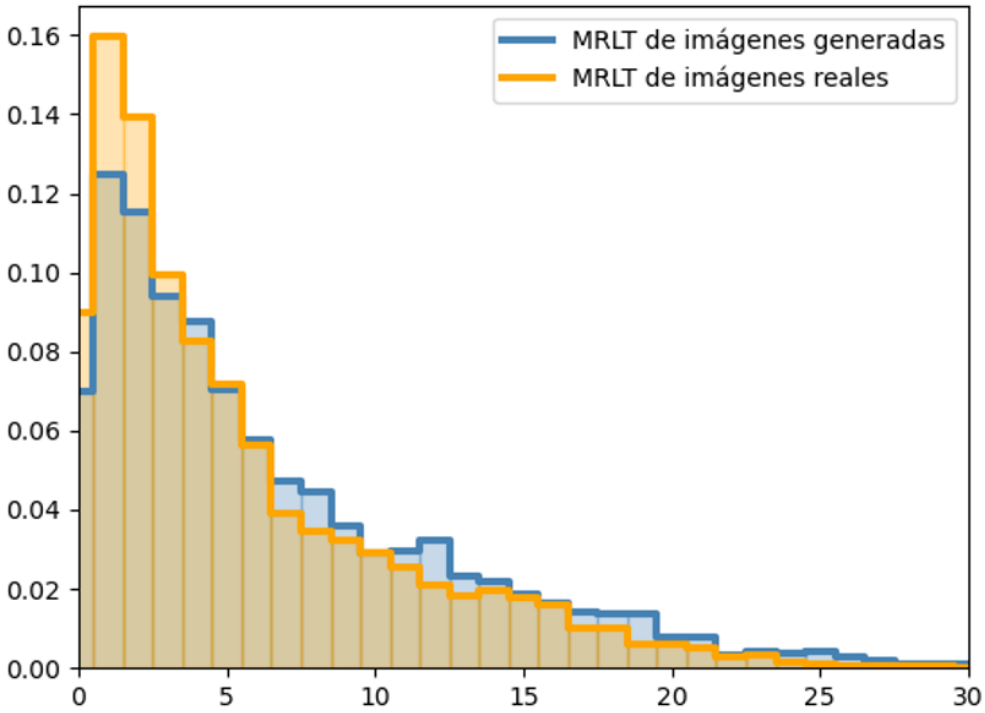

Figura 28 Gráfico con valores de MRLT de las imágenes reales y generadas de FastGAN

En los gráficos, el eje X representa los diferentes índices de los MRLT calculados y el eje Y muestra los valores de los MRLT para las imágenes generadas y para las imágenes reales. Si las barras correspondientes a las imágenes generadas son similares en altura a las barras de las imágenes reales para la mayoría de los índices, esto indica que las imágenes generadas tienen propiedades geométricas similares a las imágenes reales. Esto sugiere que la GAN está generando imágenes de calidad y diversidad similares a las del conjunto de entrenamiento.

También se tendrá en consideración el tiempo de entrenamiento necesario para alcanzar estos resultados:

Modelo	Tiempo de entrenamiento
FastGAN	28.6h
ProGAN	223.4h
StyleGAN3	172.8h

Tabla 13 Tiempo de entrenamiento de cada modelo

6. DISCUSIÓN Y ANÁLISIS DE LOS RESULTADOS

Al comparar el rendimiento entre modelos, es importante recordar primero que para evaluarlos se han utilizado métricas cuantitativas, que intentan valorar el realismo de las imágenes generadas sintéticamente. Está tarea es muy complicada y no se ha evidenciado una eficiencia absoluta, pero si ha demostrado capacidad suficiente para poder evaluar y comparar redes GAN de una manera muy útil y aproximada a la realidad. Aunque en este proyecto no se ha puesto en práctica ninguna métrica cualitativa, la observación directa de las imágenes generadas permite obtener una evaluación subjetiva rápida, y sobre todo ayudar a analizar los modelos, complementando y aportando gran valor a los resultados numéricos. Es cierto que las apreciaciones hechas basándose en la observación general de las imágenes no son aportaciones echas mediante un ensayo controlado, pero serán comentarios que se apoyarán en evidencias claras para cualquier persona que observe las imágenes.

Analizando los criterios de éxito la comparativa, el entrenamiento de cada modelo se ha realizado en condiciones idénticas utilizando un entorno local. Ha sido sencillo monitorizar el avance de los procesos, y no se ha detectado ninguna interferencia externa ni ninguna interrupción inesperada que haya podido generar algún problema durante el proceso. Las métricas se han aplicado con normalidad y los resultados obtenidos se encuentran dentro de lo esperado sin contratiempos. Todos los modelos han sido capaces de terminar generando imágenes aéreas, de mayor o menor calidad, pero son imágenes sintéticas, demostrando su capacidad para hacerlo. Y con todos los resultados obtenidos, los valores numéricos de las métricas y el tiempo de entrenamiento, junto con las imágenes generadas, proporcionan suficiente información como para realizar una evaluación a la calidad de las imágenes generadas y a la capacidad de los modelos para generar imágenes aéreas. Por estos motivos, se puede afirmar que la ejecución de la comparativa ha sido satisfactoria y fiable.

Observando los resultados conseguidos al aplicar las métricas cuantitativas de la Tabla 9, se aprecia que FastGAN y StyleGAN3 consiguen valores más positivos respecto a los de ProGAN. StyleGAN3 muestra valores ligeramente mejores que FastGAN, aunque no de manera significativamente destacable, salvo por el resultado conseguido en L2. Analizando las métricas una por una, que estos dos modelos hayan tenido un valor de FID más bajos indica

que, en términos de características de alto nivel, como son las texturas, las formas y los colores, consigue generar de imágenes de buena calidad y realismo, indicando también que ProGAN no es capaz de alcanzar esta calidad, por lo tanto, de no alcanzar el mismo realismo. Los valores altos de IS indican que las imágenes generadas por FastGAN y StyleGAN 3 son más nítidas, de mejor definición, y que muestran una variedad de características visuales, mientras que al ser el valor de ProGAN más bajo, indica que genera imágenes más borrosas, menos claras, y con repetición en los patrones visuales. El valor bajo de KID en FastGAN y StyleGAN3, indica que la distancia entre la distribución de características entre imágenes generadas y reales es pequeña, por lo que las sintéticas tienen una alta similitud con las reales utilizadas para entrenar los modelos. Al ser en ProGAN de mayor valor, se deduce que las imágenes generadas no se parecen tanto a las del dataset, lo que indica que su calidad es inferior. Al analizar L2 se puede observar que el valor de FastGAN no se asemeja al de StyleGAN3 como en el resto de las métricas, sino que se acerca más al de ProGAN. Un valor mayor de L2 significa una alta diferencia de la media cuadrática entre imágenes generadas y reales, es decir, en términos de detalles específicos concretos, contenido visual y estructuras geométricas existe una perceptible diferencia entre las imágenes de reales y sintéticas. Por lo que en este aspecto StyleGAN3 se impone frente a los otros modelos, aunque sí que hay que destacar, que la diferencia no es muy elevada respecto de la diferencia existente en las otras métricas.

En las gráficas de barras de MRLT se puede apreciar que los valores de MRLT de las imágenes reales y los valores de las imágenes generadas son bastante parecidos en cualquiera de los tres modelos. Aunque el número de barras de MRLT de imágenes generadas que superan la altura de las barras de imágenes reales es ligeramente mayor en StyleGAN3 y en FastGAN, no es una diferencia notable que implique una superioridad frente al resto. Esto se debe a que MRLT se encarga de evaluar principalmente la capacidad que tiene los generadores para para conservar la estructura geométrica esencial de las imágenes a medida que se le aplican transformaciones latentes. Al observar el dataset seleccionado para esta comparación, se nota que no mantiene una coherencia entre las formas geométricas que muestra. Al tratarse principalmente de imágenes de follaje, bosques y campos, aunque también incluye fotos con edificios, ríos y campos de cultivo. Sin embargo, estas no son la mayoría y, sobre todo, no siguen un patrón concreto de formas que los modelos puedan imitar, como si pudiera

ocurrir en un dataset de rostros de personas, por ejemplo. Por ello, el valor de MRLT de imágenes reales es tan similar al de MRLT de las imágenes generadas y, además, no se aprecia que ninguno de los modelos GAN destaque significativamente en este aspecto.

Observando las imágenes que se han generado para poder calcular todas estas métricas, algunos ejemplos se pueden observar en el Anexo B, es sencillo estar de acuerdo con los resultados que se han obtenido. StyleGAN3 y FastGAN han conseguido generar imágenes de mejor calidad, más realistas, y de mayor nitidez que ProGAN. Este último no ha conseguido alcanzar al resto, generando imágenes mucho más borrosas, donde cuesta diferenciar elementos concretos, como ríos o edificios, además de que no ha alcanzado la misma diversidad de colores y texturas, tal y como indican las métricas. StyleGAN3 y FastGAN sí han generado imágenes donde se pueden apreciar elementos concretos, aunque existe una diferencia notable en la que StyleGAN3 supera considerablemente al resto de modelos: en la generación de contornos y líneas. Aunque con FastGAN si se pueden apreciar elementos concretos como edificios, carreteras y ríos mejor que ProGAN, estos siguen siendo elementos distorsionados y poco realistas. StyleGAN3, por otro lado, es capaz de hacerlo, llegando a poder crear edificios, ríos y campos de cultivo, elementos concretos y con líneas continuas de manera más realista, aportando una calidad superior. Esta característica es especialmente relevante cuando se trata de imágenes aéreas urbanas o con elementos concretos. Lamentablemente, el dataset seleccionado, aunque incluye imágenes donde aparecen edificios y carreteras, no contiene un número significativo de imágenes urbanas o con infraestructuras humanas. Si se hubiera empleado otro dataset con un mayor número de infraestructuras humanas, como parkings o urbanizaciones, StyleGAN3 hubiera superado con mucha más diferencia a sus competidores, y FastGAN no hubiera conseguido unos resultados tan favorables.

En los tiempos empleados para entrenar los modelos, recogidos en la Tabla 10, también se pueden decir unas observaciones interesantes. Primero se puede destacar que ProGAN es el que más tiempo de entrenamiento ha requerido, y ha sido el que peores resultados ha conseguido. StyleGAN3 tiene un tiempo de entrenamiento elevado, aunque lejos del de ProGAN, y además ha conseguido resultados mejores que este, por lo que tendría concordancia. Pero FastGAN es el que más despunta, por tener un tiempo de entrenamiento mucho menor que los otros y haber conseguido resultados que, aunque no sean los mejores,

ya que StyleGAN3 ha sobresalido ligeramente, ha conseguido generar imágenes de una calidad bastante elevada. Si que hay que recordar que los entrenamientos se han detenido siguiendo un criterio subjetivo, por lo que, si el enteramiento se hubiera realizado con mayor capacidad computacional durante más tiempo, es posible que se hubieran podido apreciar mejor las diferencias entre modelos. Probablemente StyleGAN3 hubiera conseguido marcar una mayor diferencia respecto de FastGAN. Aun así, FastGAN ha demostrado conseguir resultados satisfactorios en un tiempo muy pequeño en comparación.

En resumen, las métricas han conseguido un análisis preciso del comportamiento de los modelos. ProGAN es la red GAN que ha conseguido los peores resultados. No ha conseguido un resultado positivo en ninguna de las métricas utilizadas habiendo sido superado notablemente por el resto de los modelos, algo que se puede apreciar en las imágenes generadas. StyleGAN3 es el modelo que ha conseguido los mejores resultados en todas métricas, consiguiendo generar las imágenes con mayor calidad, pero FastGAN también ha conseguido resultados de buena calidad en mucho menos tiempo. Esto significa que el factor determinante para decidir entre una u otra dependería del tiempo y la capacidad de cómputo disponible. Por lo que, si se necesita un conjunto de imágenes aéreas con urgencia, FastGAN puede conseguir resultados de un alto nivel, mientras que, si el tiempo no es un problema, lo ideal sería optar por StyleGAN3, que consigue los mejores resultados con la mejor calidad.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

En este proyecto se pretendía realizar una investigación sobre la generación de imágenes aéreas sintéticas, comparando diferentes modelos de inteligencia artificial ya existentes para determinar su capacidad para crear este tipo de imágenes. Para ello, se ha realizado un estudio de los métodos más innovadores que pudieran cumplir esta función y, entre ellos, se ha seleccionado un grupo compuesto por ProGAN, FastGAN y StyleGAN3, los cuales se han puesto en práctica entrenándose para poder generar imágenes aéreas sintéticas. Una vez conseguido esto, se han evaluado y comparado los resultados obtenidos, llegando a unas conclusiones que permiten determinar las ventajas y desventajas de cada uno de los modelos en este contexto.

Como se ha expresado durante el proyecto, disponer de conjuntos de imágenes aéreas puede ser de gran ayuda en numerosas aplicaciones prácticas, y la generación de imágenes mediante aprendizaje profundo puede contribuir a este fin de manera eficiente. Para determinar qué métodos de generación de imágenes serían los más interesantes de estudiar, se realizó una amplia investigación de todas las técnicas capaces de realizar esta tarea. Se encontraron un gran número de ellas, complicando la tarea de elegir cuáles serían las mejores, y, por supuesto, no se pudieron analizar todas en este proyecto. Aun así, se han presentado varios de estos métodos, que se encuentran en la vanguardia de la generación de imágenes, y se eligieron los tres más prometedores para el estudio.

Cuando se tuvo que decidir cómo realizar el entrenamiento de estos modelos, se intentó utilizar una plataforma en la nube, un método muy recomendable que hubiera permitido realizar el entrenamiento de manera sencilla y controlada. Sin embargo, no fue posible debido a que los servicios económicos disponibles no resultaban eficientes para este tipo de uso. Por ello, se acabaron entrenando en un ordenador de manera local, lo cual ha generado resultados igualmente válidos. Aun así, con acceso a más fondos, se podría haber realizado un entrenamiento más intensivo, utilizando un dataset mejor, mejorando las condiciones de los resultados.

De entre los datasets considerados para el estudio, se escogió uno con una resolución moderad, de 256x256 píxeles. Esta decisión fue correcta ya que las imágenes tienen

suficiente resolución para diferenciar elementos concretos y permiten reducir los tiempos de entrenamiento considerablemente. Sin embargo, un defecto del dataset es que se compone de demasiadas imágenes de zonas de bosques, árboles y campo. Aunque sí cuente con imágenes de edificios, ríos, carreteras y elementos concretos, estos no se reflejaron eficientemente en los resultados obtenidos. Por lo tanto, los resultados de este proyecto no se pueden extrapolar a cualquier tipo de imagen aérea, sino principalmente a imágenes de campos. Aun así, los resultados son útiles para hacerse una idea de cómo se comportarían los modelos con imágenes urbanas.

Para analizar los resultados se optó por métricas cuantitativas, dado que las métricas cualitativas hubieran sido más complejas de implementar. Con las métricas empleadas se ha podido evaluar los modelos de una manera muy eficiente, calculando el realismo, la calidad y la diversidad de las imágenes de manera numérica, rápida y sencilla. Dado que no existe un consenso con las métricas ideales para evaluar este tipo de modelos, se emplearon las consideradas más fiables en general, y han servido para examinar las características de los resultados, ayudando a comparar entre unos y otros.

En el análisis de los resultados de la comparativa, evaluando los resultados de las métricas, las imágenes generadas y las características recogidas, se indica claramente que el mejor modelo para generar imágenes sintéticas aéreas es SyleGAN3. Este modelo produce imágenes de mayor calidad y diversidad que sus competidores, especialmente en la generación de elementos concretos. Seguido se encuentra FastGAN que destaca por su velocidad de entrenamiento consiguiendo resultado de alta calidad en un tiempo reducido. Por último, ProGAN obtuvo valores significativamente inferiores al resto de modelos, con imágenes de menor calidad, demostrando ser el modelo menos eficaz.

Respecto a los objetivos específicos propuestos al inicio del proyecto se indica lo siguiente. Se ha profundizado en la importancia del aumento de la información sobre la generación de imágenes sintéticas y aéreas. Se han investigado los modelos generadores existentes actuales, lo que ha proporcionado la información necesaria para seleccionar los mejores algoritmos para la comparativa. Se ha preparado un entorno de entrenamiento controlado, que ha permitido ejecutar los modelos en condiciones similares, garantizando la fiabilidad de la comparación. Se ha seleccionado un dataset con imágenes aéreas de campo y bosque, permitiendo generalizar los resultados en este tipo de imágenes aéreas. Se han empleado

métricas que han evaluado la calidad y diversidad de las imágenes generadas de manera eficiente. Y se han analizado los resultados basándose en las pruebas realizadas, consiguiendo comparar los modelos y establecer las ventajas y desventajas de cada uno de ellos de manera justa y objetiva. Por lo tanto, se puede afirmar que se han cumplido con los objetivos propuestos en este proyecto.

Al comparar y analizar los resultados finales, se ha llegado a conclusiones que permiten determinar los pros y contras de cada uno de los modelos cuando se emplean para generar imágenes aéreas, pudiendo determinar cuál de ellos es mejor que los demás y por qué. En conclusión, se ha investigado los diferentes métodos de generación de imágenes y se han estudiado algunos de ellos de manera más exhaustiva, comparándolos para determinar cuál se comportaría de manera más eficiente en la generación de imágenes aéreas sintéticas. Para hacerlo, se han entrenado de manera controlada y se han evaluado los resultados obtenidos, llegando a conclusiones que permiten determinar la calidad de cada uno de ellos cuando se emplean para este objetivo. Aunque hubo complicaciones en el camino y los resultados podrían haber sido más representativos, las conclusiones obtenidas son válidas para ampliar los conocimientos en este ámbito, mejorando la investigación sobre el comportamiento de los modelos en la generación de imágenes aéreas y cumpliendo con los objetivos establecidos en el proyecto.

7.2. Líneas de trabajo futuro

Como líneas de trabajo a futuro para futuras investigaciones y proyectos:

- Cambiar el dataset utilizando conjuntos de datos que representen todo tipo de imágenes aéreas: urbanizaciones, campos, ríos, puertos, playas, etc. De esta manera se conseguirá un resultado que consiga una representación más generalizada de las imágenes aéreas, aunque para conseguirlo se requerirá de una mayor capacidad computacional.
- Aplicando otras métricas además de las que se han empleado se podrían llegar a deducir más detalles sobre las características de cada uno de los modelos. Además, se podría realizar un análisis cuantitativo de los resultados, lo que mejoraría mucho el análisis.

- En este proyecto se ha hablado sobre otros modelos que podrían servir para este objetivo y que no se han llegado a entrenar, como por ejemplo StyleGAN2, VAE o DCGAN, y existen un gran número de ellos, tantos que sería imposible nombrarlos todos. Sería interesante comparar otros modelos y ver también como se comportan.
- En este proyecto se han comparado con el objetivo de generar imágenes aéreas, pero también se podría realizar la comparación con otros objetivos, como para generar rostros u objetos. De esta manera se comprobaría también la flexibilidad de los modelos.

Referencias bibliográficas

Alibani, M., Acito, N., & Corsini, G. (2024). Multispectral satellite image generation using StyleGAN3. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Anantrasirichai, N., Biggs, J., Albino, F., & Bull, D. (2019). A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. Remote Sensing of Environment, 230, 111179.

Atenas, F., Sanhueza, F., & Valenzuela, C. Redes Neuronales Adversarias Convolucionales para Generación de Imágenes.

Audebert, N., Le Saux, B., & Lefèvre, S. (2019). Deep learning for classification of hyperspectral data: A comparative review. IEEE geoscience and remote sensing magazine, 7(2), 159-173.

Barisic, A., Petric, F., & Bogdan, S. (2022). Sim2air-synthetic aerial dataset for uav monitoring. IEEE Robotics and Automation Letters, 7(2), 3757-3764.

Bellovin, S. M., Dutta, P. K., & Reitinger, N. (2019). Privacy and synthetic datasets. Stan. Tech. L. Rev., 22, 1.

Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying mmd gans. arXiv preprint arXiv:1801.01401.

Borji, A. (2019). Pros and cons of gan evaluation measures. Computer vision and image understanding, 179, 41-65.

Brownlee J. (2019, julio 12) How to Evaluate Generative Adversarial Networks. *Machine Learning Mastery.* https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/

Choudhary, A., Verma, P. K., & Rai, P. (2022, December). Comparative study of various cloud service providers: A review. In 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS) (pp. 1-8). IEEE.

Clarcat. (2024, mayo 23). *Comparativa: Amazon Web Services (AWS) vs. Microsoft Azure vs. Google Cloud Platform*. https://www.clarcat.com/comparativa-aws-vs-microsoft-azure-vs-google-cloud-platform/

DataWolke (2023, marzo 17) Comparación de las soluciones de Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson [Publicación]. LinkedIn. Recuperado el 23 de mayo de 2024 de https://www.linkedin.com/pulse/comparaci%C3%B3n-de-las-soluciones-machine-learning-service-amazon/

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., ... & Raskar, R. (2018). Deepglobe 2018: A challenge to parse the earth through satellite images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 172-181).

Deng, X., Zhu, Y., & Newsam, S. (2019). Using conditional generative adversarial networks to generate ground-level views from overhead imagery. arXiv preprint arXiv:1902.06923.

Fortet, R., & Mourier, E. (1953). Convergence de la répartition empirique vers la répartition théorique. In Annales scientifiques de l'École Normale Supérieure (Vol. 70, No. 3, pp. 267-285).

Ganguli, S., Garzon, P., & Glaser, N. (2019). GeoGAN: A conditional GAN with reconstruction and style loss to generate standard layer of maps from satellite images. arXiv preprint arXiv:1902.05611.

Gao, F., Yang, Y., Wang, J., Sun, J., Yang, E., & Zhou, H. (2018). A deep convolutional generative adversarial network (DCGANs)-based semi-supervised method for object recognition in synthetic aperture radar (SAR) images. Remote Sensing, 10(6), 846.

Gao, Q., Shen, X., & Niu, W. (2020). Large-scale synthetic urban dataset for aerial scene understanding. IEEE Access, 8, 42131-42140.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

Huang, H., He, R., Sun, Z., & Tan, T. (2018). Introvae: Introspective variational autoencoders for photographic image synthesis. Advances in neural information processing systems, 31.

Hutchinson, B., Rostamzadeh, N., Greer, C., Heller, K., & Prabhakaran, V. (2022, June). Evaluation gaps in machine learning practice. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 1859-1876).

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr.

K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu and J. Jiang, "Edge-Enhanced GAN for Remote Sensing Image Superresolution," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 8, pp. 5799-5812, Aug. 2019, doi: 10.1109/TGRS.2019.2902431.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. Advances in neural information processing systems, 34, 852-863.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8110-8119).

Khrulkov, V., & Oseledets, I. (2018, July). Geometry score: A method for comparing generative adversarial networks. In International conference on machine learning (pp. 2621-2629). PMLR.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Lamata, L. (2021, January). Quantum reinforcement learning with quantum photonics. In Photonics (Vol. 8, No. 2, p. 33). MDPI.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016, June). Autoencoding beyond pixels using a learned similarity metric. In International conference on machine learning (pp. 1558-1566). PMLR.

Little, C., Elliot, M., Allmendinger, R., & Samani, S. S. (2021). Generative adversarial networks for synthetic data generation: a comparative study. arXiv preprint arXiv:2112.01925.

Liu, B., Zhu, Y., Song, K., & Elgammal, A. (2020, October). Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In International Conference on Learning Representations.

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017, July). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In 2017 IEEE International geoscience and remote sensing symposium (IGARSS) (pp. 3226-3229). IEEE.

Man, K., & Chahl, J. (2022). A review of synthetic image data and its use in computer vision. Journal of Imaging, 8(11), 310.

Man, K., & Chahl, J. (2022). A review of synthetic image data and its use in computer vision. Journal of Imaging, 8(11), 310.

Manjaly, S. (2022, julio 26). Amazon AWS vs. Microsoft Azure vs. Google Cloud: ¿qué proveedor en la nube es mejor? *Invgate.* https://blog.invgate.com/es/amazon-aws-vs-microsoft-azure-vs-google-cloud

Marra, F., Gragnaniello, D., Cozzolino, D., & Verdoliva, L. (2018, April). Detection of gan-generated fake images over social networks. In 2018 IEEE conference on multimedia information processing and retrieval (MIPR) (pp. 384-389). IEEE.

Mino, A., & Spanakis, G. (2018, December). Logan: Generating logos with a generative adversarial neural network conditioned on color. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 965-970). IEEE.

My Grafic Card (2024, junio 2024) *TPU Vs GPU: Unraveling the Differences and Performance.* *https://mygraphicscard.com/tpu-vs-gpu/?utm_content=cmp-true*

Página de Amazon Mechanical Turk (https://www.mturk.com/)

Página de Apache Spark (https://spark.apache.org/)

Página de AWS (https://docs.aws.amazon.com/)

Página de Google Cloud (https://cloud.google.com/docs?hl=es-419)

Página de IBM (https://www.ibm.com/docs/en)

Página de Microsoft Azure (https://learn.microsoft.com/en-us/azure/?product=popular)

Página de Nvidia (https://www.nvidia.com/en-us/data-center/)

Página de Unreal Engine (https://www.unrealengine.com/de/spotlights/unreal-studio-brings-cityengine-neighborhood-to-life)

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., & Kim, Y. (2018). Data synthesis based on generative adversarial networks. arXiv preprint arXiv:1806.03384.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Rampini, L., & Re Cecconi, F. (2024). Synthetic images generation for semantic understanding in facility management. Construction Innovation, 24(1), 33-48.

Raul Martynek. (2023, septiembre 22). Nvidia vs. The Cloud Providers [Artículo]. LinkedIn. Recuperado el 23 de mayo de 2024 de https://www.linkedin.com/pulse/nvidia-vs-cloud-providers-raul-martynek/

Razavi, A., Van den Oord, A., & Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems, 32.

Repositorio de FastGAN (https://github.com/odegeasslbc/FastGAN-pytorch)

Repositorio de FID, IS y KID (https://github.com/toshas/torch-fidelity)

Repositorio de Geometric Score (https://github.com/KhrulkovV/geometry-score)

Repositorio de ProGAN (https://github.com/facebookresearch/pytorch_GAN_zoo)

Repositorio de StyleGAN 3(https://github.com/NVlabs/stylegan3)

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.

Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions, and future directions. ACM Computing Surveys (CSUR), 54(3), 1-42.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singh, P., & Komodakis, N. (2018, July). Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium (pp. 1772-1775). IEEE.

Świrski, L., & Dodgson, N. (2014, March). Rendering synthetic ground truth images for eye tracker evaluation. In Proceedings of the Symposium on Eye Tracking Research and Applications (pp. 219-222).

Tan, Y. F., Loo, J. Y., Ting, C. M., Noman, F., Phan, R. C. W., & Ombao, H. (2024, April). BrainFC-CGAN: A Conditional Generative Adversarial Network for Brain Functional Connectivity Augmentation and Aging Synthesis. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1511-1515). IEEE.

Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. Advances in neural information processing systems, 30.

Vargas Orellana, A. D. (2024). Future Perspectives in Image Generation: Advancements in GANs and QGANs.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., & Schmid, C. (2017). Learning from synthetic humans. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 109-117).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., & Shotton, J. (2021). Fake it till you make it: face analysis in the wild using synthetic data alone. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3681-3691).

Xia, G. S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., ... & Lu, X. (2017). AID: A benchmark data set for performance evaluation of aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 55(7), 3965-3981.

Xu, L., & Veeramachaneni, K. (2018). Synthesizing tabular data using generative adversarial networks. arXiv preprint arXiv:1811.11264.

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. Advances in neural information processing systems, 32.

Yang, J., Kannan, A., Batra, D., & Parikh, D. (2017). Lr-gan: Layered recursive generative adversarial networks for image generation. arXiv preprint arXiv:1703.01560.

Yates, M., Hart, G., Houghton, R., Torres, M. T., & Pound, M. (2022). Evaluation of synthetic aerial imagery using unconditional generative adversarial networks. ISPRS Journal of Photogrammetry and Remote Sensing, 190, 231-251.

Zhao, B., Zhang, S., Xu, C., Sun, Y., & Deng, C. (2021). Deep fake geography? When geospatial data encounter Artificial Intelligence. Cartography and Geographic Information Science, 48(4), 338–352. https://doi.org/10.1080/15230406.2021.1910075

Zhao, S., Liu, Z., Lin, J., Zhu, J. Y., & Han, S. (2020). Differentiable augmentation for data-efficient gan training. Advances in neural information processing systems, 33, 7559-7570.

Zhao, Z., Kunar, A., Birke, R., & Chen, L. Y. (2021, November). Ctab-gan: Effective table data synthesizing. In Asian Conference on Machine Learning (pp. 97-112). PMLR.

Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Zhang, W., ... & Wang, J. (2017). Activation maximization generative adversarial nets. arXiv preprint arXiv:1703.02000.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

Anexo A. Ejemplos de imágenes del dataset

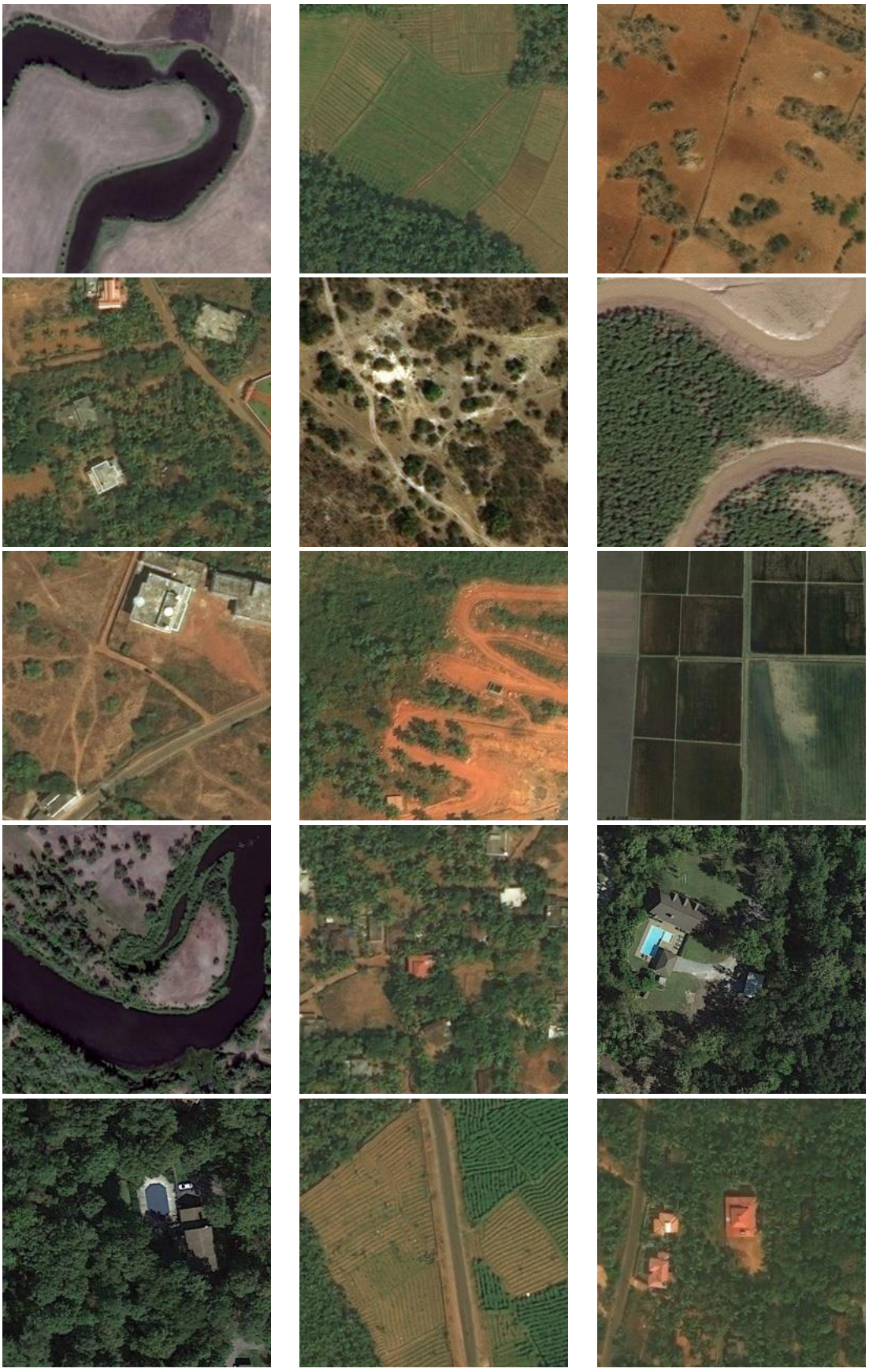

Figura 29 Imágenes aleatorias del dataset Fuente: Demir, I. et al., 2018

Anexo B. Ejemplos de imágenes generadas

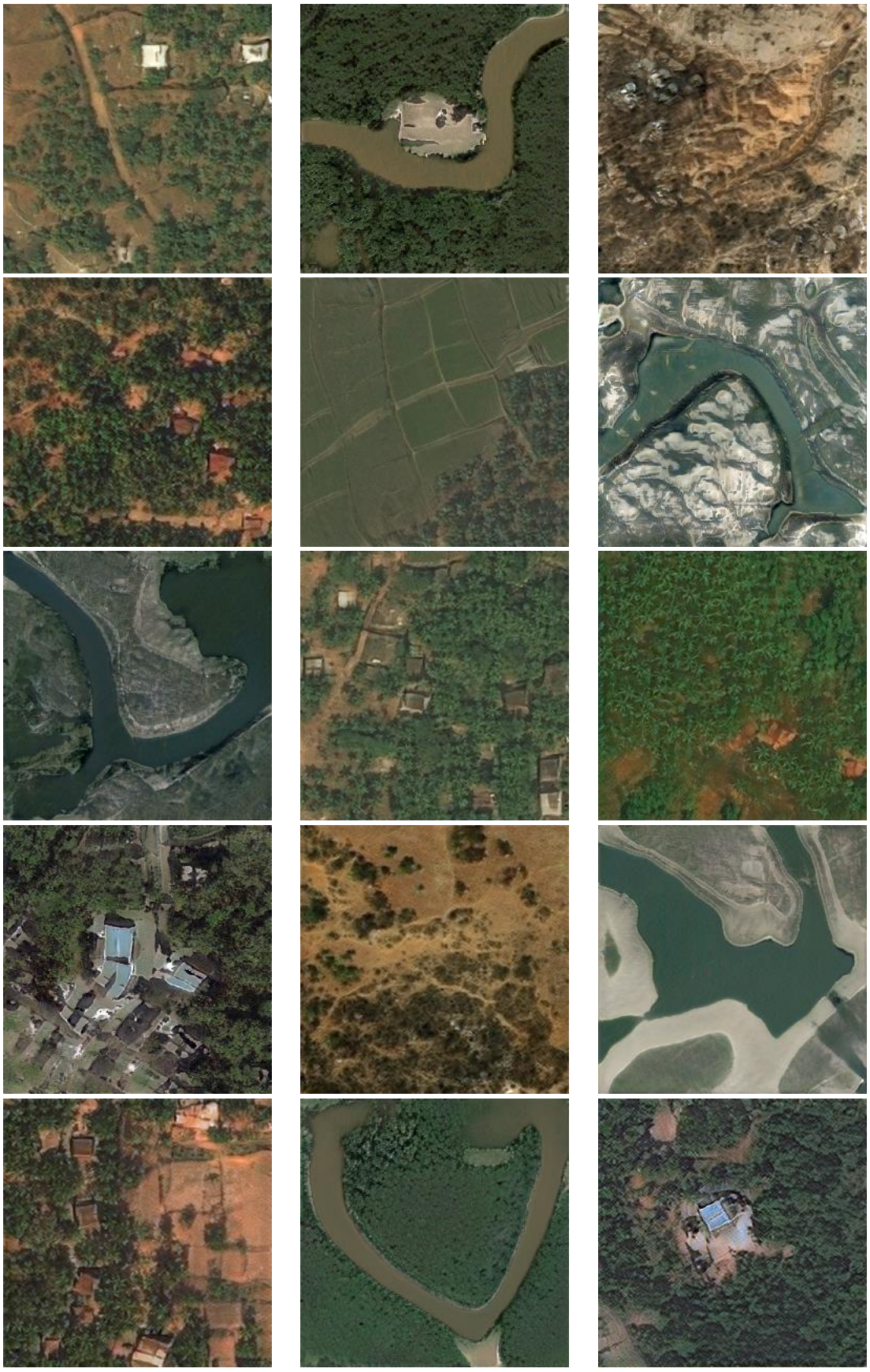

Figura 30 Imágenes aleatorias de generadas con FastGAN

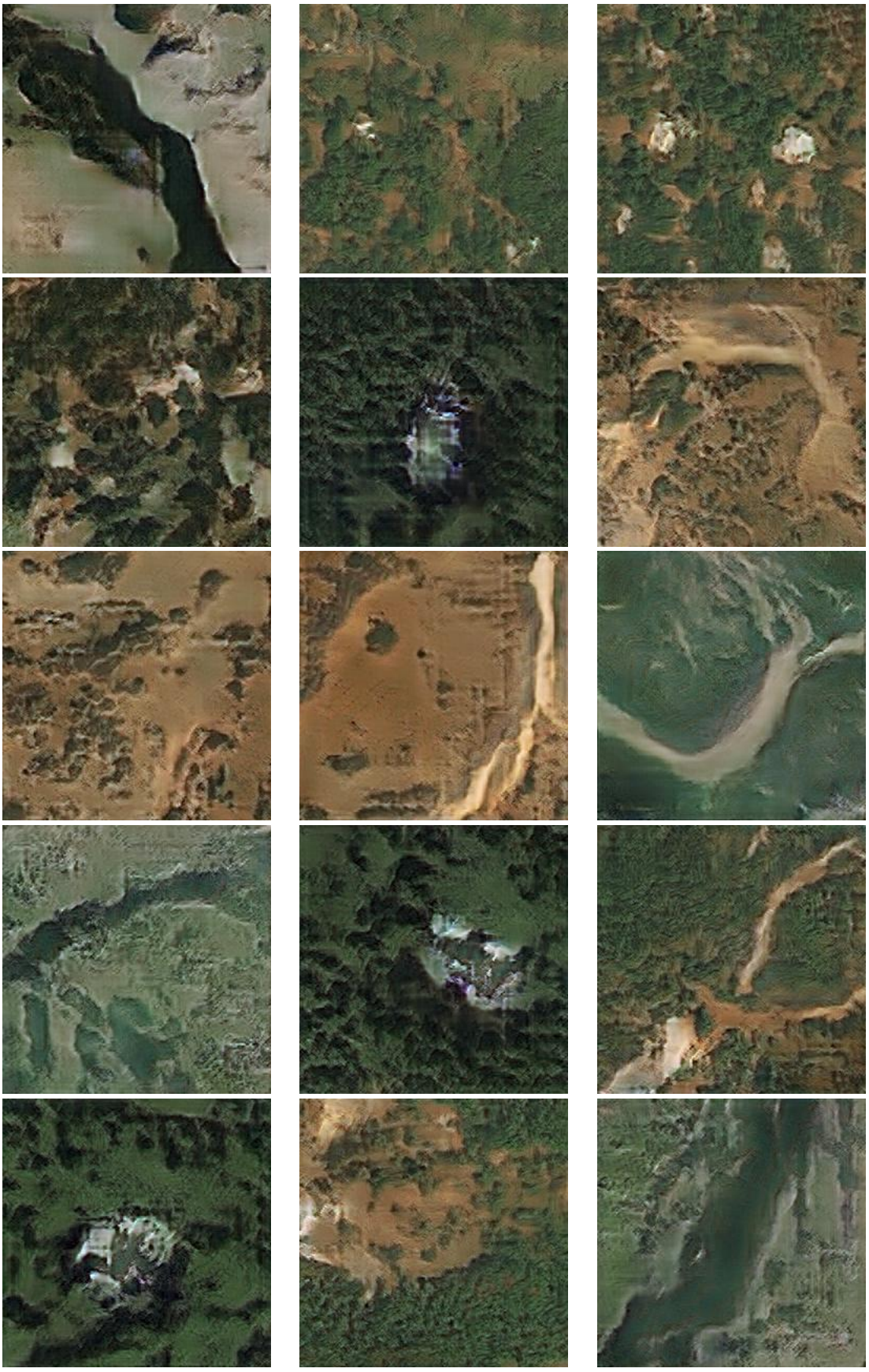

Figura 31 Imágenes aleatorias de generadas con ProGAN

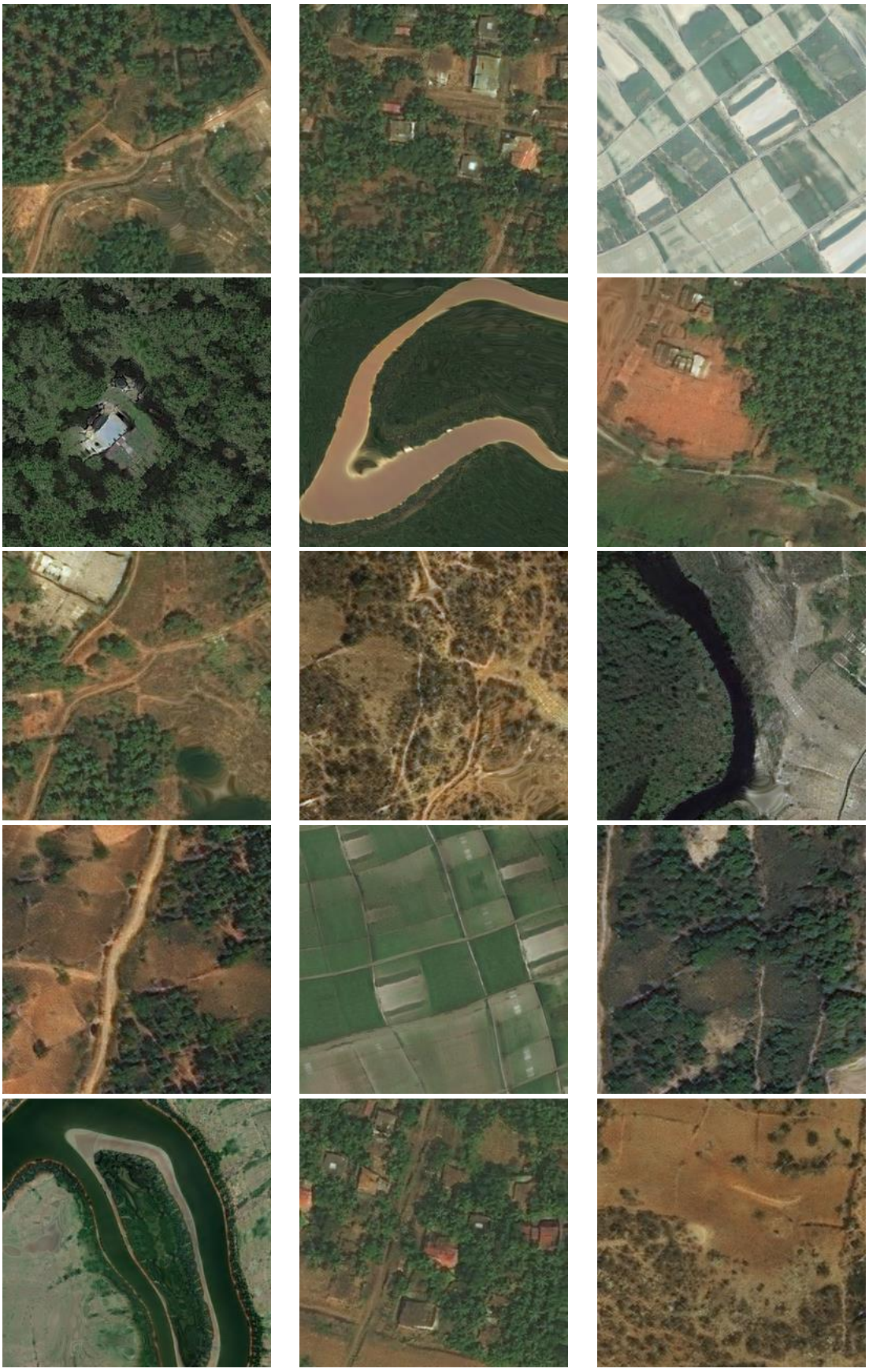

Figura 32 Imágenes aleatorias de generadas con Stylegan3

Anexo C. Código empleado

En el siguiente repositorio se encuentran todos los códigos que se han empleado para el desarrollo de esta comparativa:

Repositorio GitHub del proyecto: https://github.com/guillerdm/Comparativa-de-algoritmos-de-generaci-n-de-data-sets-sint-ticos-para-vistas-a-reas