



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Modelos generativos de imágenes de objetos en vuelo

Autor: Raúl Gómez Téllez

Grado: Ingeniería en Informática

Directores: Cristóbal José Carmona del Jesús y Pedro González García

Departamento del profesor: Departamento de Informática

Fecha: 26/06/2025

Licencia CC



Agradecimientos

La verdad es que la redacción de esta parte ha sido la más difícil. Hay tantas personas que me han apoyado a lo largo de este proyecto, que podría escribir páginas y páginas. Sin embargo, intentaré ser lo más conciso posible para expresar mi gratitud a todas aquellas sin las cuales este proyecto no habría sido posible.

En primer lugar, quiero agradecer a mis tutores, Cristóbal y Pedro, por darme la oportunidad de realizar tanto este proyecto como las prácticas de empresa en el DaS-Cl, y por permitirme formar parte del grupo SIMIDAT.

Hablando de SIMIDAT, no puedo olvidarme de mis compañeros: A Nono, nuestro experto solucionador de problemas. A David, por enseñarme que las representaciones del conocimiento no son tan abstractas como parecen, y por los paseos terapéuticos a la fuente para llenar la botella. A Fernando, por aportar ese toque de humor en los momentos de desesperación. A Mariasún, por las visitas al despacho cargadas de entusiasmo por trabajar. Y a Manu, por resolver todas mis dudas sobre lo que se podía hacer y lo que no.

También tengo que agradecer a mis amigos, por acompañarme durante todo este proceso, animándome constantemente y siendo los primeros evaluadores (y críticos) de lo “bien” que salían las imágenes en las primeras fases.

Por último, pero lo más importante, quiero agradecer a mis padres, las personas más importantes de mi vida. Mamá, gracias por tu creatividad y ese toque de alegría constante que me has transmitido. Papá, gracias por ser un manitas y por haberme contagiado desde pequeño ese gusto por cacharrear con todo.

Sé que criarme no ha sido la tarea más fácil del mundo, y que ha estado llena de emociones de todo tipo: desde discusiones por cosas triviales hasta abrazos, risas y muchísimas experiencias felices. Puede que no seamos una familia perfecta, ni tengamos grandes lujos, pero tengo todo lo que necesito. No cambiaría esta vida por nada.

Muchas gracias. Os quiero un montón.

Tabla de contenidos

1. INTRODUCCIÓN	3
1.1. Motivación	3
1.2. Objetivos	4
1.3. Metodología y planificación	5
1.3.1. Metodología	5
1.3.2. Planificación temporal	5
1.4. Herramientas y Recursos	6
1.4.1. Software	6
1.4.2. Hardware	7
1.4.3. Presupuesto	7
2. ANTECEDENTES	11
2.1. Introducción	11
2.2. Contexto	12
2.3. Inteligencia Artificial y Ciencia de Datos	13
2.3.1. Fundamentos de la Inteligencia Artificial	13
2.3.2. Explosión del aprendizaje profundo y modelos generativos	15
2.3.3. Jerarquía y especialización en la IA moderna	16

2.3.4. Evolución reciente en visión por computador	16
2.3.5. Fundamentos de redes neuronales artificiales	18
2.3.6. Ciencia de Datos: fundamentos y evolución	19
2.4. Generación de imágenes	22
2.4.1. Pipeline conceptual	23
2.4.2. Aumentación de datos	23
2.4.3. Técnicas clásicas	23
2.4.4. Aumentación basada en modelos generativos	24
2.4.5. Riesgos y buenas prácticas	24
2.4.6. Transferencia de dominio	24
2.4.7. Sobreajuste a datos sintéticos	26
2.4.8. Generación condicionada	28
2.5. Evaluación y métricas	28
2.5.1. Métricas objetivas	29
2.5.2. Evaluación subjetiva	30
2.5.3. Resumen comparativo de métricas	30
2.5.4. Criterio adoptado en este trabajo	31
2.6. Consideraciones éticas y de uso responsable	32
2.6.1. Posibles riesgos	32
2.7. Conexión con el proyecto	33
3. Modelos Generativos	37
3.1. Introducción	37
3.2. Modelos generativos: panorámica	38

3.2.1. Evolución histórica	39
3.2.2. Tres enfoques predominantes	39
3.2.3. Comparación cualitativa	39
3.3. Autoencoders Variacionales (VAE)	40
3.3.1. Fundamentos teóricos	40
3.3.2. Arquitectura genérica	41
3.3.3. Implementación básica en PyTorch	42
3.3.4. Aplicaciones a UAVs	42
3.3.5. Principales variantes del VAE	42
3.3.6. Valoración del enfoque VAE	48
3.4. Redes Generativas Adversariales (GAN)	48
3.4.1. Fundamentos teóricos	48
3.4.2. Arquitectura genérica	49
3.4.3. Implementación básica en PyTorch	50
3.4.4. Aplicaciones a UAVs	50
3.4.5. Principales variantes	51
3.4.6. Valoración del enfoque GAN	57
3.5. Modelos de Difusión (DDPM)	57
3.5.1. Fundamentos teóricos	57
3.5.2. Arquitectura genérica de DDPM	58
3.5.3. Aplicaciones a UAVs	59
3.5.4. Implementación básica en PyTorch	59
3.5.5. Variantes principales de modelos de difusión	60
3.5.6. Valoración del enfoque modelos de difusión	66

3.6. Comparativa de las arquitecturas generativas	67
4. Metodología y Experimentación	69
4.1. Introducción	69
4.2. Visión general del flujo de trabajo	70
4.3. Diseño experimental	70
4.3.1. Criterios de selección de modelos	71
4.3.2. Hipótesis de trabajo	71
4.3.3. Variables y configuración experimental	72
4.3.4. Esquema del flujo de trabajo	72
4.4. Recursos computacionales y herramientas	72
4.4.1. Hardware utilizado	73
4.4.2. Entornos de desarrollo	74
4.4.3. Gestión de experimentos	74
4.4.4. Datos	75
4.4.5. Entorno de desarrollo y bibliotecas	77
4.4.6. Control de versiones y documentación	78
4.5. Modelos seleccionados	78
4.5.1. Modelos GAN	79
4.5.2. Modelos VAE	79
4.5.3. Modelos de Difusión	80
4.5.4. Resumen de modelos	80
4.5.5. Procedimiento experimental	81
5. Resultados	83

5.1. Metodología de evaluación	84
5.1.1. Métricas utilizadas	84
5.1.2. Procedimiento experimental	85
5.1.3. Evaluación cualitativa	85
5.1.4. Consideraciones y limitaciones	85
5.1.5. Síntesis comparativa	86
5.2. Reconstrucción en CelebA con modelos VAE	86
5.2.1. Evaluación cuantitativa	87
5.2.2. Análisis visual	87
5.2.3. Conclusión parcial	92
5.3. Reconstrucción en UAV con modelos VAE	93
5.3.1. Evaluación cuantitativa	93
5.3.2. Análisis visual	93
5.3.3. Análisis de resultados	98
5.4. Generación en CelebA con GANs y modelos de difusión	99
5.4.1. Evaluación cuantitativa	99
5.4.2. Análisis visual	99
5.4.3. Relación arquitectura–imagen	103
5.4.4. Análisis de resultados	103
5.5. Generación en UAV con GANs y modelos de difusión	104
5.5.1. Evaluación cuantitativa	104
5.5.2. Análisis visual	104
5.5.3. Relación arquitectura–imagen	112
5.5.4. Análisis de resultados	113

5.6. Discusión comparativa y síntesis	113
5.6.1. Arquitectura y rendimiento: VAE, GAN y FDM	114
5.6.2. Tarea: reconstrucción vs. generación	115
5.6.3. Dominio: CelebA vs. UAV	115
5.6.4. Reflexión final	116
6. Resumen y conclusiones	119
6.1. Resumen general del trabajo	119
6.2. Lecciones aprendidas	120
6.3. Líneas futuras de investigación	122
6.4. Conclusión Final	123
Anexo 1: Recursos utilizados	125
Bibliografía	132

Lista de figuras

1.1.	Cronograma del TFG.	6
2.1.	Jerarquía conceptual de subcampos de la inteligencia artificial. Inspira- do en @thatshelbs.	17
2.2.	Ciclo iterativo típico del proceso de Ciencia de Datos. Fuente:[1]	21
2.3.	Intersección de disciplinas necesarias para la Ciencia de Datos.Fuente [2]	22
2.4.	Esquema simplificado de un sistema de generación de imágenes. La condición <i>c</i> es opcional.	23
2.5.	Representación esquemática de la brecha de dominio entre datos sin- téticos y reales.Elaboración propia	26
3.1.	Línea temporal de los hitos clave en modelos generativos profundos (2014–2024). Elaboración propia	37
3.2.	Hitos abreviados en la evolución de los modelos generativos profundos. .	39
3.3.	Esquema general de un Autoencoder Variacional (VAE), con codifica- ción, muestreo y decodificación. Elaboración propia	41
3.4.	Esquema básico de una GAN. Elaboración propia	49
3.5.	Esquema simplificado de un paso de denoising en DDPM. Elaboración propia	58
4.1.	Flujo de trabajo general: de la revisión bibliográfica al análisis de resul- tados.	70

4.2. Esquema general del flujo de trabajo experimental. Elaboración propia	72
4.3. Collage comparativo. Izquierda: subconjunto de CelebA. Derecha: subconjunto del dataset propio de drones.	77
5.1. Imágenes reales del conjunto CelebA utilizadas como referencia.	88
5.2. Reconstrucciones generadas por el modelo WAE-MMD en CelebA.	89
5.3. Reconstrucciones generadas por el modelo Info-vae en CelebA.	90
5.4. Reconstrucciones generadas por el modelo β -vae en CelebA.	91
5.5. Imágenes reales del conjunto UAV.	94
5.6. Reconstrucciones generadas por el modelo WAE-MMD en UAV.	95
5.7. Reconstrucciones generadas por el modelo WAE-MMD en UAV.	96
5.8. Reconstrucciones generadas por el mode lo β -vae en UAV.	97
5.9. Imágenes reales del conjunto CelebA.	100
5.10. Imágenes generadas por StyleGAN2-ADA en CelebA.	101
5.11. Imágenes generadas por FDM-EDM en CelebA.	102
5.12. Imágenes reales del conjunto UAV.	105
5.13. Imágenes generadas por Stylegan-2 en UAV.	106
5.14. Imágenes generadas por DCGAN en UAV.	107
5.15. Imágenes generadas por WGAN en UAV.	108
5.16. Imágenes generadas por FDM-EDM en UAV.	109
5.17. Imágenes generadas por FDM-VP en UAV.	110
5.18. Imágenes generadas por FDM-EP en UAV.	111
5.19. Comparativa visual de resultados generativos en UAV: imagen real (izquierda), StyleGAN2-ADA, FDM-EDM y WAE-MMD (reconstrucción). Se aprecian diferencias en textura, definición y coherencia estructural. .	116

5.20. Comparativa visual de resultados generativos en Celeb-A: imagen real (izquierda), StyleGAN2-ADA, FDM-EDM y WAE-MMD (reconstrucción). Se aprecian diferencias en textura, definición y coherencia estructural. . 116

Lista de tablas

1.1. Tareas del TFG por fases y objetivos.	6
1.2. Estimación de costes asociados al proyecto en un entorno profesional. Los valores se han obtenido a partir de tarifas públicas actualizadas de AWS, Lambda Labs, CNMC y portales salariales de IA en 2024–2025.	8
2.1. Técnicas comunes de transferencia de dominio en entornos con datos sintéticos.	27
2.2. Comparativa de métricas utilizadas para evaluar la calidad generativa.	31
3.1. Visión de alto nivel de las fortalezas y debilidades de cada familia.	39
3.2. Comparativa general entre VAE, GAN y DDPM.	67
4.1. Modelos generativos seleccionados para la experimentación.	80
5.1. Resultados de reconstrucción con VAE en CelebA.	87
5.2. Resultados de reconstrucción con VAE en el conjunto UAV.	93
5.3. Resultados de generación en el conjunto CelebA.	99
5.4. Resultados de generación en el conjunto de drones (UAV).	104
5.5. Comparativa global entre modelos generativos evaluados en CelebA y UAV, diferenciando por arquitectura, tarea y conjunto. Las celdas con “–” indican métricas no aplicables en generación pura.	114

Lista de listados de código

3.1. Implementación básica de un VAE en PyTorch	42
3.2. Implementación básica de una GAN en PyTorch	50
3.3. Implementación básica de un modelo DDPM	59

Resumen

La generación sintética de imágenes mediante modelos generativos profundos se ha convertido en una herramienta clave para la ampliación de conjuntos de datos en tareas de visión por computador. Este Trabajo Fin de Grado se centra en la evaluación y comparación de diferentes arquitecturas generativas —incluyendo Autoencoders Variacionales (VAE), Redes Generativas Adversariales (GAN) y Modelos de Difusión (DDPM)— aplicadas a la generación y reconstrucción de imágenes de objetos voladores no tripulados (UAVs).

El trabajo combina un análisis teórico riguroso con un enfoque experimental, utilizando dos conjuntos de datos: el benchmark facial CelebA y un conjunto propio de imágenes de UAVs recopilado y aumentado a partir de fuentes públicas. Se evalúa el rendimiento de cada arquitectura mediante métricas cuantitativas estándar (FID, IS, PSNR, SSIM) y un análisis cualitativo visual, considerando tanto fidelidad como diversidad. Asimismo, se estudia el impacto de la generación sintética en la generalización de modelos discriminativos entrenados con estos datos.

Los resultados muestran cómo los modelos de difusión superan en calidad visual a GAN y VAE, a costa de un mayor tiempo de inferencia, mientras que los VAE ofrecen ventajas en términos de control latente e interpretabilidad. El documento concluye con una reflexión sobre los riesgos del sobreajuste a datos sintéticos, la necesidad de técnicas de transferencia de dominio, y propone líneas futuras centradas en distilación de modelos y generación condicionada. Este trabajo aporta una base experimental sólida para la integración de técnicas generativas en sistemas de percepción autónoma.

Capítulo 1

INTRODUCCIÓN

1.1. Motivación

En apenas una década, los modelos generativos han pasado de ser un artefacto de laboratorio a convertirse en el motor de buena parte de la innovación en visión por computador, síntesis de datos y creación de contenido. Algunos de los los hitos más representativos incluyen: la aparición de las *Redes Generativas Adversariales* (GAN) en 2014 [3], los *Autoencoders Variacionales* (VAE) en 2013–2014 [4, 5], hasta la irrupción de los *Modelos de Difusión* y sus variantes latentes entre 2020 y 2024 [6, 7, 8]. Cada avance ha venido acompañado de mejoras en estabilidad, calidad y control, así como de nuevas aplicaciones que abarcan desde el diseño asistido hasta la medicina.

La evolución de estas arquitecturas ha convertido a la generación sintética en una herramienta clave para enfrentar limitaciones frecuentes en el acceso a datos reales. En particular, en tareas como detección, segmentación o clasificación visual, la capacidad de generar imágenes artificiales permite simular escenarios complejos, aumentar la diversidad del dataset y reducir el sesgo.

En este contexto, surge el presente trabajo, centrado en la **detección y seguimiento de vehículos aéreos no tripulados (UAVs)**. La obtención de imágenes reales de UAVs es costosa y limitada por múltiples factores: condiciones de vuelo, restricciones legales, privacidad o simplemente disponibilidad técnica. Contar con imágenes sintéticas realistas —donde el dron aparece en diferentes altitudes, perspectivas y fondos— resulta crucial para entrenar modelos robustos capaces de generalizar en condiciones reales.

Además, los modelos generativos ofrecen la posibilidad de controlar atributos semánticos, como el tipo de dron, el ángulo de visión o las condiciones de iluminación, lo que permite generar casos de borde y situaciones difíciles de capturar *in situ*. Esta capacidad de generación controlada es particularmente valiosa en sistemas de seguridad, análisis táctico o entrenamiento de redes de visión por computador en entornos adversos.

Con este objetivo, se plantea el análisis comparativo de tres familias generativas que han marcado la evolución reciente del campo: los **Autoencoders Variacionales (VAE)**, las **Redes Generativas Adversariales (GAN)** y los **Modelos de Difusión (DDPM)**. Cada una de estas aproximaciones será evaluada en términos de fidelidad visual, control semántico y aplicabilidad al dominio UAV, sentando así las bases para futuras líneas de mejora y adaptación en contextos reales.

1.2. Objetivos

Esta sección define el objetivo general que guía el desarrollo del presente Trabajo de Fin de Grado, así como los objetivos específicos que permiten alcanzarlo. El objetivo principal consiste en **investigar y evaluar distintas arquitecturas generativas para la creación sintética de imágenes de objetos en vuelo**, con el fin de identificar qué enfoques ofrecen un mejor equilibrio entre calidad visual, diversidad de resultados y viabilidad computacional.

Para alcanzar el objetivo principal del trabajo, se definen los siguientes objetivos específicos

- Realizar una revisión bibliográfica exhaustiva sobre modelos fundacionales para la generación sintética de imágenes.
- Analizar y diseñar distintos modelos generativos adaptados a la tarea específica de generar imágenes de objetos en vuelo.
- Desarrollar un estudio experimental para evaluar la efectividad y calidad de los modelos generativos implementados.
- Comparar los resultados obtenidos y establecer recomendaciones sobre las técnicas más adecuadas para la generación de imágenes sintéticas en este dominio.

1.3. Metodología y planificación

1.3.1. Metodología

La metodología adoptada en este trabajo combina un enfoque sistemático con una vertiente experimental. El proceso se estructura en las siguientes etapas:

1. Revisión detallada del estado del arte en técnicas de generación de imágenes sintéticas, con especial énfasis en modelos GAN, VAE y DDPM.
2. Estudio y aplicación de algoritmos metaheurísticos y estrategias de optimización orientadas al ajuste de los modelos generativos.
3. Diseño y ajuste de diferentes arquitecturas adaptadas a la generación de imágenes de objetos en vuelo.
4. Implementación práctica de los modelos seleccionados mediante frameworks de aprendizaje profundo.
5. Ejecución de experimentos controlados para evaluar el rendimiento de los modelos en términos de calidad, diversidad y realismo.
6. Análisis crítico de los resultados obtenidos y elaboración de conclusiones técnicas con base en métricas cuantitativas.

1.3.2. Planificación temporal

La planificación del Trabajo Fin de Grado se ha estructurado en torno a seis fases principales: investigación, implementación, experimentación y documentación transversal. Cada fase se ha desglosado en tareas específicas asociadas a un objetivo particular, con fechas de inicio y fin claramente delimitadas. Esta planificación permite distribuir el trabajo de forma equilibrada a lo largo del cuatrimestre, asegurando una progresión lógica desde la exploración teórica hasta la evaluación práctica y la documentación final del proyecto.

La Tabla 1.1 resume las tareas contempladas, su relación con los objetivos definidos y su duración aproximada. La documentación se ha considerado una tarea transversal, activa desde las primeras fases de investigación hasta la entrega final, con el fin de garantizar la trazabilidad y calidad del desarrollo.

El desarrollo de cada tarea ha sido concebido en paralelo siempre que ha sido posible, buscando la eficiencia en el uso del tiempo y la superposición de bloques compatibles. La Figura 1.1 muestra el diagrama de Gantt resultante, donde se visualiza la distribución temporal de las tareas y su interdependencia.

Fase	Objetivo	Tarea	Inicio	Fin
Investigación	1	Revisión modelos	2025-02-01	2025-03-15
Investigación	2	Metaheurísticas	2025-02-20	2025-03-30
Implementación	3	Recopilación arquitecturas	2025-04-01	2025-04-15
Implementación	4	Implementación modelos	2025-04-10	2025-05-05
Experimentación	5	Evaluación	2025-05-06	2025-05-25
Experimentación	6	Análisis final	2025-05-20	2025-06-10
Documentación (transversal)	1–6	Documentación	2025-02-10	2025-06-25

Tabla. 1.1: Tareas del TFG por fases y objetivos.

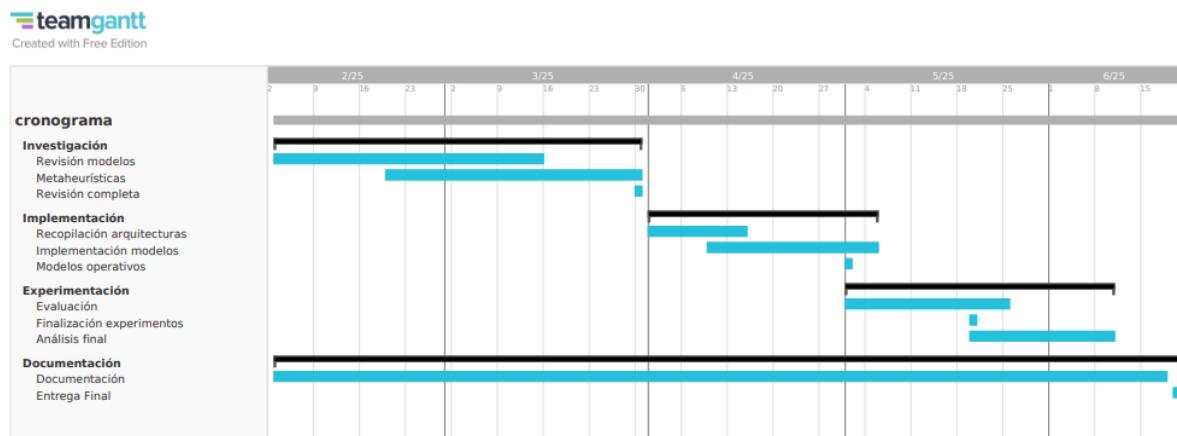


Figura 1.1: Cronograma del TFG.

1.4. Herramientas y Recursos

En esta sección se detallan las herramientas y recursos utilizados durante las diferentes fases de este trabajo, desde la investigación hasta el desarrollo y evaluación de los modelos generativos.

1.4.1. Software

- **Lenguaje de programación:** Python fue el lenguaje elegido por su robustez y su amplia comunidad en el ámbito del aprendizaje automático.

- **Frameworks de aprendizaje profundo:** Se emplearon *PyTorch* y *TensorFlow* para la implementación y entrenamiento de los modelos generativos, incluyendo modelos GAN, VAE y DDPM.
- **Control de versiones:** Se utilizó *Git* para gestionar el código fuente y mantener un historial organizado durante todo el desarrollo.
- **Investigación y análisis bibliográfico:** Durante la fase de revisión, se utilizaron gestores de referencias como *Zotero*, junto con bases de datos académicas como *Google Scholar*, *IEEE Xplore*, *Elsevier* y *arXiv*.

1.4.2. Hardware

- **Servidor GPU principal:** Se contó con acceso a un sistema *NVIDIA DGX-1*, equipado con 8 GPUs Tesla V100, que permitió entrenar modelos complejos de forma eficiente.
- **Alternativa con GPUs individuales:** Se llevaron a cabo pruebas y entrenamientos preliminares en dos máquinas locales, equipadas con GPUs *NVIDIA GeForce RTX 4060 Ti* y *RTX 3070*, respectivamente.
- **Almacenamiento:** Se dispuso de almacenamiento tanto local como en red para la gestión de los datasets y la persistencia de resultados experimentales.

1.4.3. Presupuesto

Aunque el desarrollo de este proyecto se ha realizado principalmente con recursos propios y del entorno académico, a continuación se presenta una estimación de los costes reales que implicaría su ejecución en un entorno profesional, considerando tanto costes humanos como computacionales:

- **Costes humanos:** Se estima que el tiempo dedicado (aproximadamente 8 meses a tiempo parcial, equivalentes a 600 horas) tendría un coste de mercado en una empresa del sector IA entre 25–35 €/h¹. Se adopta una media de 30 €/h, lo que supone un coste total aproximado de 18 000 €.
- **Costes computacionales:** Se contemplan tres modalidades:

¹Según informes de remuneración de Glassdoor y Talent.com para perfiles junior de IA en España en 2024.

- **DGX-1 (8 GPUs Tesla V100):** Plataformas como Runpod.io, Lambda Labs o instancias dedicadas de GCP y AWS estiman un coste horario entre 35–50 €/h². Para 200 horas, esto implica entre 7 000 y 10 000 €.
- **GPU local (RTX 4060 Ti / RTX 3070):** Con un consumo energético medio de 200 W por GPU y un precio de electricidad doméstica de 0,15 €/kWh³, el coste energético para 200 horas es de unos 12 €. A esto se añade una amortización por uso y mantenimiento estimada en 0,5 €/h⁴. El coste total se eleva así a un rango de 600–1 000 € en hardware propio.
- **Almacenamiento en la nube:** Un volumen de 1 TB en GCP/AWS/S3 (acceso estándar, región Europa) tiene un coste mensual estimado entre 10–20 €⁵. Para una duración de proyecto de hasta 6 meses, esto representa entre 120 y 250 €.
- **Software:** No se incurrió en licencias de pago durante el proyecto. Sin embargo, en un entorno profesional podría requerirse software adicional (editores, herramientas de seguimiento, control de versiones privado, visores de IA, etc.), que no ha sido contemplado en esta estimación.
- **Otros gastos:** Incluyen impresión, encuadernación y presentación del documento final. Se estima un coste puntual de 30 €, habitual en entornos universitarios.

Concepto	Estimación mínima	Estimación máxima
Costes de personal	18 000€	18 000€
Alquiler de infraestructura tipo DGX-1 (200 h @ 35–50 €/h)	7 000 €	10 000 €
Uso de GPUs RTX 4060 Ti / 3070 en local (200 h + electricidad)	600 €	1 000 €
Almacenamiento en la nube (1 TB en GCP / AWS)	120 €	250 €
Software	0€	0€
Otros gastos	30€	30€
Gastos totales	25 750€	29 280€

Tabla. 1.2: Estimación de costes asociados al proyecto en un entorno profesional. Los valores se han obtenido a partir de tarifas públicas actualizadas de AWS, Lambda Labs, CNMC y portales salariales de IA en 2024–2025.

El presente presupuesto ha sido ajustado para reflejar de forma realista el coste de un entorno profesional de desarrollo y evaluación de modelos generativos. En particular, se ha considerado el uso de infraestructura certificada como las estaciones DGX-1 de NVIDIA, con múltiples GPUs V100, o instancias equivalentes en plataformas en la

²Costes basados en tarifas de instancias p3.16xlarge en AWS y A100/V100 en Lambda Labs a fecha de mayo 2025.

³Según la CNMC (Comisión Nacional de los Mercados y la Competencia) para el precio medio de electricidad en hogares españoles, 2025.

⁴Estimación propia basada en vida útil media de 3 años de uso semiprofesional intensivo.

⁵Fuente: calculadora oficial de Google Cloud Storage y AWS Simple Monthly Calculator.

nube como Google Cloud Platform (GCP) o Amazon Web Services (AWS). En dichos entornos, el coste horario de una instancia de alto rendimiento con 8 GPUs puede oscilar entre 35 y 50 €/h, lo que justifica la elevación del coste estimado para 200 horas de computación intensiva.

Además, se ha incluido un coste mayor para almacenamiento en la nube (hasta 1 TB) y se han contemplado posibles licencias profesionales de software (como herramientas de edición, visualización o aceleradores de IA). Este ajuste responde a los estándares de costes reales observados en proyectos que siguen buenas prácticas según normativas como ISO/IEC 27001, que implican infraestructuras seguras, trazabilidad y supervisión profesional de los recursos computacionales empleados.

Este enfoque garantiza una estimación más realista, transparente y alineada con las condiciones necesarias para reproducibilidad, fiabilidad y escalabilidad de experimentos en entornos académicos o industriales.

Capítulo 2

ANTECEDENTES

2.1. Introducción

El auge reciente de la inteligencia artificial (IA) ha impulsado la generación de imágenes sintéticas hasta niveles de realismo inimaginables hace apenas una década [9]. Este capítulo reúne los principios teóricos que sustentan dicho avance y los contextualiza en la problemática concreta que aborda este trabajo: la síntesis de imágenes de objetos en vuelo —principalmente vehículos aéreos no tripulados (UAV)— orientada al apoyo de tareas de detección y análisis.

La exposición se estructura de lo general a lo particular. Se parte de los conceptos fundamentales de la IA y el *machine learning*, para posteriormente conectar con el aprendizaje profundo y las técnicas de descubrimiento de conocimiento (*Knowledge Discovery in Databases*, KDD), que permiten extraer valor de grandes volúmenes de datos. A continuación, se abordan las estrategias clásicas y modernas de aumentación de datos, se introduce la motivación técnica tras la generación automática de imágenes, y se ofrece una panorámica de las tres familias de modelos generativos que vertebran el resto del documento: Redes Generativas Adversariales (GAN), Autocodificadores Variacionales (VAE) y Modelos de Difusión (DDPM). El capítulo se completa con una síntesis de las métricas de evaluación más utilizadas, y un apartado dedicado a las consideraciones éticas y al uso responsable de estas tecnologías.

Este recorrido proporciona al lector el marco conceptual necesario para comprender los desarrollos técnicos del Capítulo 3, la metodología experimental descrita en el Capítulo 4 y la discusión de resultados expuesta en los capítulos finales. De este modo, los antecedentes actúan como hilo conductor entre los fundamentos teóricos

generales y su aplicación específica a la generación sintética de imágenes de UAV, que constituye el núcleo de esta investigación.

2.2. Contexto

La última década ha sido testigo de una expansión vertiginosa de la inteligencia artificial (IA) aplicada a la visión por computador, impulsada por la disponibilidad de grandes bases de datos visuales y el incremento sostenido de la potencia computacional en GPUs [10]. Hitos como AlexNet en 2012 [11] marcaron un punto de inflexión al demostrar que las redes neuronales profundas podían superar con amplitud los métodos clásicos en tareas de clasificación de imágenes. Desde entonces, la investigación se ha diversificado hacia desafíos más complejos —detección, segmentación, seguimiento y, de manera muy relevante para este trabajo, generación sintética— abriendo nuevas oportunidades en sectores tan dispares como la medicina, el entretenimiento o la defensa.

Dentro de este espectro, el dominio de los objetos en vuelo presenta singularidades que incrementan su dificultad: variaciones extremas de escala y ángulo, fondos dinámicos y condiciones de iluminación muy variables. Para aplicaciones de vigilancia y seguridad, el reconocimiento de vehículos aéreos no tripulados (UAV) se ha convertido en una prioridad estratégica. Sin embargo, los conjuntos de datos públicos disponibles para esta tarea son escasos y, en muchos casos, carecen de anotaciones exhaustivas o variabilidad suficiente [12]. En este escenario, la generación de imágenes sintéticas realistas —que respeten la cinemática y las texturas propias de estas aeronaves— se plantea como una solución eficaz para:

- **Aumentar la diversidad de entrenamiento:** incorporar variaciones de postura, altitud y fondo difíciles o imposibles de capturar en condiciones controladas.
- **Reducir costes y riesgos:** evitar campañas de filmación con UAV reales, las cuales suelen ser costosas y estar sujetas a restricciones legales o logísticas.
- **Facilitar la evaluación robusta:** generar escenarios adversos (*edge cases*) que pongan a prueba la resiliencia de los sistemas de detección.

En este contexto, los modelos generativos profundos se posicionan como una herramienta clave para suplir la escasez de datos reales y acelerar el desarrollo iterativo

de sistemas de visión por computador, tanto embarcados como terrestres. Los capítulos siguientes profundizarán en las técnicas que hacen posible esta síntesis, así como en los experimentos diseñados para cuantificar su impacto en la detección de UAV.

2.3. Inteligencia Artificial y Ciencia de Datos

La Inteligencia Artificial (IA) y la Ciencia de Datos constituyen los pilares teóricos y metodológicos sobre los que se construyen las técnicas modernas de generación de imágenes sintéticas. Esta sección ofrece un recorrido estructurado por los conceptos clave, su evolución histórica y su interrelación práctica. Se analizan los fundamentos de la IA, el auge del aprendizaje automático y profundo, así como los principios de la Ciencia de Datos y su papel en el diseño de sistemas inteligentes basados en datos. El objetivo es proporcionar el marco necesario para comprender el contexto en el que se desarrollan y aplican los modelos generativos analizados en este trabajo.

2.3.1. Fundamentos de la Inteligencia Artificial

La noción de *inteligencia* ha sido históricamente esquiva. Desde las definiciones psicométricas que enfatizan el razonamiento lógico [13] hasta los enfoques de inteligencias múltiples propuestos por Gardner [14], la literatura coincide en un rasgo común: la capacidad de adaptarse a entornos cambiantes resolviendo problemas de manera eficaz. Trasladar esa aptitud al ámbito computacional constituye la ambición fundacional de la inteligencia artificial (IA).

Definición operativa.

En el contexto de sistemas computacionales, Russell y Norvig definen la IA como «la construcción de agentes que perciben su entorno y actúan razonablemente para maximizar su probabilidad de éxito» [15]. Esta noción articula percepción, razonamiento y acción, permitiendo un marco formal para el desarrollo de algoritmos adaptativos.

Orígenes históricos y transición estadística

El concepto de una máquina capaz de razonar fue anticipado mucho antes del nacimiento de los ordenadores modernos. En 1950, Alan Turing propuso su famoso "juego de imitación", hoy conocido como el *test de Turing*, como criterio operativo para determinar si una máquina puede considerarse inteligente [16]. Este test no evalúa el funcionamiento interno de la máquina, sino su comportamiento observado: si un interlocutor humano no puede distinguir entre la conversación con una persona y una máquina, esta última puede considerarse inteligente. Este enfoque conductual sentó las bases para una evaluación funcionalista de la inteligencia artificial.

El término *Inteligencia Artificial* fue acuñado formalmente por John McCarthy en 1956 durante la conferencia de Dartmouth, evento que se considera el punto fundacional de la IA como campo académico. Aquella primera etapa de entusiasmo apostó por métodos simbólicos: sistemas expertos, motores de inferencia y técnicas de lógica formal que representaban el conocimiento mediante reglas explícitas. Estos sistemas mostraron un rendimiento prometedor en contextos cerrados, pero fracasaban sistemáticamente al enfrentarse a la ambigüedad, incertidumbre o complejidad del mundo real.

Durante las décadas de 1960 y 1970, la investigación en IA generó grandes expectativas —impulsadas por avances como el programa ELIZA o el sistema experto DENDRAL—, pero pronto se evidenciaron sus limitaciones. La imposibilidad de capturar todo el conocimiento en reglas formales, junto con el alto coste computacional de las inferencias lógicas, llevó a una sucesión de *inviernos de la IA*: períodos de drástica reducción en la financiación y el interés institucional debido al estancamiento de los resultados.

Hubo al menos dos inviernos claramente identificables. El primero se produjo en la década de 1970, cuando el optimismo inicial se desvaneció al no cumplirse las expectativas prometidas en tareas como el procesamiento del lenguaje natural y la traducción automática. El segundo gran invierno se dio a finales de los años 80 y principios de los 90, cuando los sistemas expertos, muy costosos de mantener y escasamente escalables, demostraron ser insostenibles frente a aplicaciones reales cambiantes. Ambos períodos supusieron una advertencia sobre los límites del enfoque simbólico y la necesidad de modelos más adaptativos, lo que preparó el terreno para la irrupción de métodos basados en datos.

El resurgimiento llegó a partir de los años 80 con el auge del *aprendizaje automático* (machine learning), una aproximación estadística que sustituía las reglas programadas

das por el ajuste de parámetros a partir de datos. En lugar de diseñar explícitamente el conocimiento, el sistema lo "aprende" detectando patrones en los ejemplos observados. Métodos como los árboles de decisión, las máquinas de soporte vectorial (SVM), y los primeros modelos bayesianos permitieron abordar tareas más complejas, desde clasificación médica hasta sistemas de recomendación. Esta etapa marcó un cambio de paradigma: del razonamiento simbólico al inductivo, sentando las bases del enfoque actual basado en datos.

2.3.2. Explosión del aprendizaje profundo y modelos generativos

A pesar de que las redes neuronales artificiales habían sido propuestas ya en los años 50 (como el perceptrón de Rosenblatt), su uso práctico fue durante mucho tiempo limitado por la falta de datos, poder computacional y técnicas de entrenamiento eficientes. No fue hasta 2006, con la introducción del concepto de *preentrenamiento capa a capa* por Hinton et al., que se comenzaron a entrenar redes profundas con mejores resultados. Sin embargo, el punto de inflexión llegó en 2012, con la aparición de AlexNet [11], una red neuronal convolucional profunda que ganó el concurso ImageNet con una ventaja abrumadora frente a métodos tradicionales. Este éxito demostró que, con suficientes datos y capacidad computacional (GPUs), las redes profundas podían superar ampliamente a los métodos clásicos en tareas visuales complejas.

Desde entonces, el *aprendizaje profundo* (deep learning) se ha convertido en el paradigma dominante en IA. Arquitecturas como VGG, ResNet, Inception o DenseNet han permitido escalar la profundidad de los modelos, mientras que nuevas técnicas como la normalización por lotes (batch normalization), el dropout o los optimizadores como Adam han mejorado la estabilidad del entrenamiento. El deep learning no solo ha revolucionado la visión por computador, sino también el procesamiento del lenguaje natural, la generación de audio, la síntesis de texto y, muy especialmente, la generación de imágenes.

En este contexto surgen los **modelos generativos**, que no solo predicen etiquetas o regresan valores, sino que son capaces de crear contenido completamente nuevo. Las **Redes Generativas Adversariales** (GAN), introducidas por Goodfellow et al. en 2014 [3], proponen un esquema competitivo entre dos redes que aprenden a generar imágenes realistas. Posteriormente, los **Autoencoders Variacionales** (VAE) [4] y los **Modelos de Difusión** [6] han complementado este enfoque con mecanismos probabilísticos y estocásticos, abriendo nuevas posibilidades en calidad y control de la generación.

Esta revolución ha tenido un impacto profundo no solo en la investigación académica, sino también en la industria: desde asistentes virtuales y sistemas de recomendación, hasta herramientas de diseño, edición automática y generación de contenido multimedia. La capacidad de crear datos sintéticos útiles ha generado un nuevo paradigma en el entrenamiento de sistemas de visión artificial, como los estudiados en este trabajo.

2.3.3. Jerarquía y especialización en la IA moderna

A medida que la inteligencia artificial ha evolucionado, ha dado lugar a una serie de subdisciplinas jerárquicamente estructuradas, cada una con un enfoque más específico y técnicas más especializadas. Esta estructura refleja tanto el desarrollo histórico como la diversidad metodológica que caracteriza al campo. La Figura 2.1 resume esta jerarquía conceptual:

- **IA:** engloba cualquier sistema diseñado para emular capacidades cognitivas humanas como el razonamiento, la planificación o el aprendizaje.
- **Aprendizaje automático (Machine Learning, ML):** subcampo que emplea algoritmos capaces de identificar patrones en los datos y aprender sin ser explícitamente programados para cada tarea.
- **Aprendizaje profundo (Deep Learning):** rama del ML que utiliza redes neuronales profundas para aprender representaciones jerárquicas directamente desde los datos sin necesidad de ingeniería manual.
- **IA generativa:** área dedicada a modelos que generan nuevo contenido en lugar de simplemente analizar el existente; abarca imágenes, texto, música y más.
- **Modelos generativos específicos:** como GAN, VAE y modelos de difusión, que representan enfoques particulares dentro de la IA generativa, con énfasis en calidad visual, control semántico y eficiencia de entrenamiento.

2.3.4. Evolución reciente en visión por computador

En el ámbito de la visión por computador, el progreso ha sido especialmente acelerado en la última década. Antes de la adopción generalizada del aprendizaje profundo,

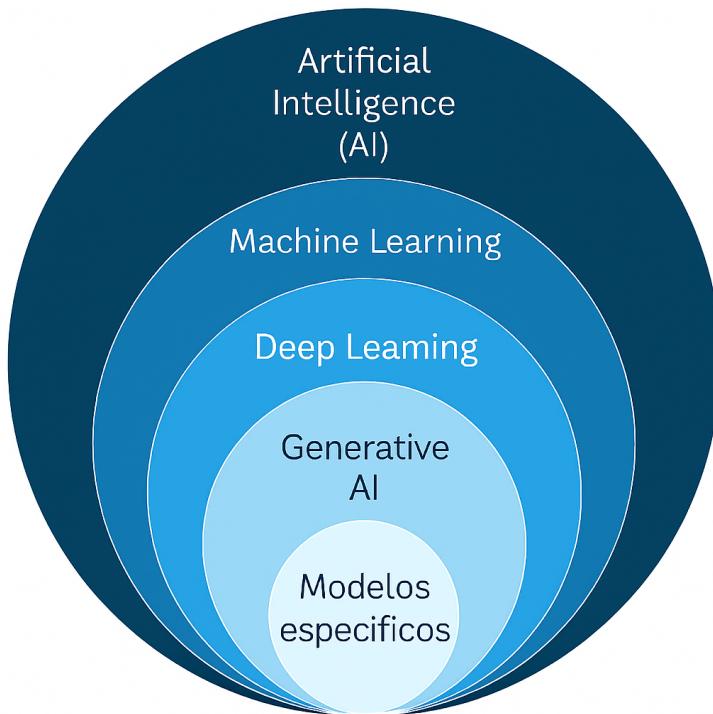


Figura 2.1: Jerarquía conceptual de subcampos de la inteligencia artificial. Inspirado en [@thatshelbs](#).

los sistemas de visión se basaban en técnicas manuales de extracción de características como SIFT (Scale-Invariant Feature Transform), HOG (Histogram of Oriented Gradients) o LBP (Local Binary Patterns). Estas técnicas, aunque eficaces en entornos controlados, requerían una importante intervención humana para definir y seleccionar descriptores relevantes, lo que limitaba su aplicabilidad a tareas complejas o escenarios no estructurados.

El año 2012 marcó un punto de inflexión con la aparición de AlexNet [11], una red neuronal convolucional profunda que revolucionó el campo al ganar el desafío ImageNet con una ventaja significativa respecto a los métodos tradicionales. AlexNet demostró que el uso de arquitecturas profundas entrenadas con GPUs y grandes cantidades de datos podía superar con creces a los métodos manuales, abriendo el camino a una nueva generación de modelos más precisos y escalables.

Desde entonces, se han sucedido avances notables. Modelos como VGGNet, ResNet o DenseNet han mejorado la capacidad de generalización y profundidad de las redes. En paralelo, se han desarrollado modelos especializados para tareas concretas: YOLO (You Only Look Once) [17] y SSD (Single Shot Multibox Detector) permiten la detección de objetos en tiempo real con alta precisión, mientras que U-Net [18] se ha convertido en el estándar de facto en segmentación semántica, especialmente en entornos biomédicos.

Más recientemente, los *transformers visuales* como ViT (Vision Transformer) [19] han introducido una arquitectura alternativa basada en mecanismos de atención auto-regresiva, originalmente concebidos para procesamiento de lenguaje. Estos modelos han demostrado una capacidad de aprendizaje comparable —e incluso superior— a las CNNs tradicionales, especialmente cuando se entrenan en grandes volúmenes de datos.

En conjunto, la evolución de la visión por computador ha pasado de enfoques estáticos y manuales a sistemas de aprendizaje profundo altamente adaptativos, que constituyen hoy el núcleo de aplicaciones como la detección de UAVs, el reconocimiento facial, la conducción autónoma o la inspección automatizada en entornos industriales.

2.3.5. Fundamentos de redes neuronales artificiales

Las redes neuronales artificiales (RNA) constituyen la piedra angular del aprendizaje profundo. Inspiradas en el funcionamiento del cerebro humano, estas redes están compuestas por unidades mínimas denominadas neuronas artificiales, que realizan una operación matemática básica: una combinación lineal de las entradas seguida de una transformación no lineal mediante una función de activación. La operación típica de una neurona es:

$$y = \sigma \left(\sum_i w_i x_i + b \right)$$

donde x_i son las entradas, w_i los pesos asociados, b un término de sesgo, y σ una función de activación como ReLU, tanh o sigmoid.

Estas neuronas se organizan en capas, formando una red estructurada. Las principales capas incluyen:

- **Capas densas (fully connected):** cada neurona se conecta con todas las de la capa anterior. Son comunes en redes clásicas y en las capas finales de clasificadores.
- **Capas convolucionales (CNN):** diseñadas para procesar imágenes, utilizan filtros que detectan patrones espaciales como bordes o texturas.

- **Capas recurrentes (RNN)**: adecuadas para secuencias temporales; almacenan información previa mediante estados ocultos. Se emplean en tareas como análisis de trayectorias o predicción temporal.
- **Capas de atención (transformers)**: ponderan dinámicamente la importancia de cada entrada, permitiendo una mayor capacidad de modelado contextual. Han ganado protagonismo en visión y lenguaje.

Durante el entrenamiento, las redes ajustan sus parámetros internos mediante el algoritmo de retropropagación (backpropagation), que calcula los gradientes de una función de pérdida respecto a los pesos, y los actualiza usando optimizadores como SGD, RMSprop o Adam. Este proceso se repite iterativamente hasta minimizar el error.

Las redes neuronales profundas (deep neural networks) contienen múltiples capas ocultas, lo que les permite aprender representaciones jerárquicas: las primeras capas capturan rasgos simples como bordes o colores, mientras que las últimas abstraen conceptos más complejos como formas, patrones o categorías semánticas.

Esta capacidad de abstracción progresiva ha hecho de las RNA una herramienta indispensable para tareas como clasificación de imágenes, detección de objetos, segmentación semántica y, en el caso de este trabajo, la generación sintética de imágenes UAV.

2.3.6. Ciencia de Datos: fundamentos y evolución

La Ciencia de Datos (*Data Science*) se ha consolidado como una disciplina clave en la era digital, con el objetivo de extraer conocimiento útil a partir de grandes volúmenes de datos. Si bien su consolidación como campo independiente es reciente, sus raíces se encuentran en disciplinas tradicionales como la estadística, la minería de datos y la informática científica. Esta intersección ha permitido abordar problemas complejos desde una perspectiva práctica, basada en el análisis empírico, la automatización del aprendizaje y la interpretación de resultados.

Definición operativa. La Ciencia de Datos puede definirse como el conjunto de metodologías, herramientas y principios orientados a convertir datos brutos en información estructurada, conocimiento útil y decisiones accionables. Va más allá del análisis estadístico: implica limpieza, modelado, validación y comunicación de resultados, todo ello articulado mediante un ciclo iterativo de mejora continua.

KDD: la raíz formal. Un antecedente directo del enfoque actual es el paradigma del Descubrimiento de Conocimiento en Bases de Datos (KDD, por sus siglas en inglés), formalizado en los años 90 por Fayyad et al. [20]. Este modelo establece un proceso compuesto por varias etapas secuenciales:

- Selección de datos relevantes.
- Limpieza y preprocesamiento.
- Transformación e ingeniería de características.
- Minería de datos (aplicación de algoritmos para identificación de patrones).
- Evaluación e interpretación de los hallazgos.

Este esquema sentó las bases para lo que hoy se conoce como el ciclo de vida de la Ciencia de Datos.

El ciclo de vida en Ciencia de Datos

Actualmente, la Ciencia de Datos se concibe como un proceso iterativo que abarca desde la formulación del problema hasta la generación de valor aplicado. El ciclo más comúnmente aceptado incluye:

1. **Definición del problema:** comprensión del contexto y objetivos analíticos.
2. **Recolección de datos:** recopilación desde múltiples fuentes (sensores, APIs, bases SQL, imágenes).
3. **Preparación:** limpieza, tratamiento de valores nulos, normalización.
4. **Análisis exploratorio:** visualizaciones, estadísticas descriptivas, correlaciones.
5. **Modelado:** entrenamiento de modelos predictivos, clasificadores o agrupamientos.
6. **Evaluación:** comparación con métricas apropiadas (accuracy, F1, MSE, etc.).
7. **Despliegue y comunicación:** informes, dashboards, visualización de resultados.

Este flujo se representa esquemáticamente en la Figura 2.2, donde se muestra el carácter cíclico del proceso y la interdependencia entre etapas. Lejos de ser una secuencia rígida, este ciclo permite iterar en función de los hallazgos en cada fase, lo que hace de la Ciencia de Datos una disciplina adaptativa por naturaleza.

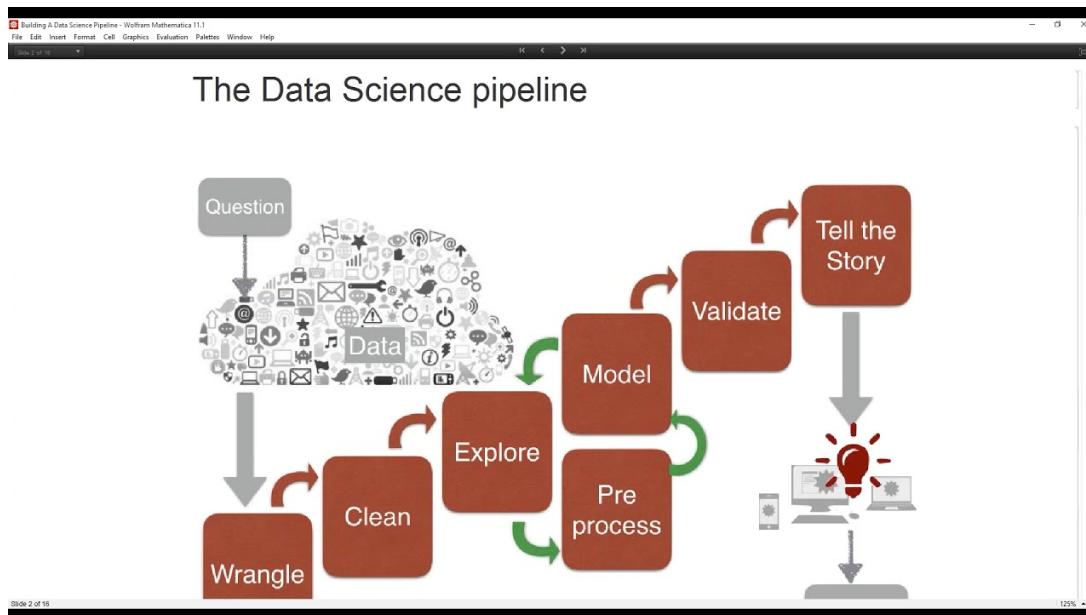


Figura 2.2: Ciclo iterativo típico del proceso de Ciencia de Datos. Fuente:[1]

Este ciclo permite iterar entre etapas cuando los resultados no son satisfactorios, incorporando retroalimentación continua, lo que hace de la Ciencia de Datos un enfoque adaptativo por naturaleza.

La figura del científico de datos

La evolución de esta disciplina ha generado una nueva figura profesional: el científico de datos. Esta persona opera en la intersección de tres grandes dominios (ver Figura 2.3): conocimientos del dominio de aplicación, habilidades estadísticas y experiencia en programación. Esta versatilidad le permite formular preguntas pertinentes, construir modelos y traducir los resultados en acciones concretas.

Además de su base técnica, el científico de datos debe ser capaz de comunicar hallazgos de forma clara y orientada a la toma de decisiones, actuando como puente entre los datos y los responsables del negocio o del diseño experimental.

Así, la Ciencia de Datos se configura como una disciplina estratégica, indispensable para el análisis y modelado en dominios complejos como el reconocimiento visual de UAVs, la generación de datos sintéticos o el entrenamiento de modelos robustos,

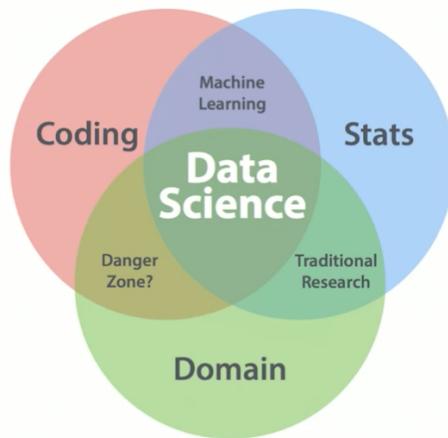


Figura 2.3: Intersección de disciplinas necesarias para la Ciencia de Datos. Fuente [2]

que se exploran a lo largo del presente trabajo.

2.4. Generación de imágenes

La síntesis automática de imágenes persigue aprender una función G capaz de transformar un vector latente $z \sim \mathcal{N}(0, I)$ —o su versión condicionada z, c — en una imagen $x \in \mathbb{R}^{H \times W \times C}$ tal que la distribución resultante $p_G(x)$ sea indistinguible de la distribución de datos reales $p_{\text{data}}(x)$ [3]. Más allá de constituir un reto intelectual, esta capacidad habilita aplicaciones de alto impacto:

- **Aumentación de datos:** abordada en la Sección 2.4.2, permite reducir el sobreajuste en dominios con escasez de muestras.
- **Simulación y entrenamiento:** posibilita generar entornos fotorrealistas para probar algoritmos de navegación o seguimiento sin riesgos físicos.
- **Reconstrucción e *inpainting*:** rellena regiones perdidas o permite la expansión de resolución (superresolución) [21].
- **Síntesis controlada:** genera imágenes coherentes con descripciones textuales, mapas de segmentación o parámetros físicos (*conditional generation*).

2.4.1. Pipeline conceptual

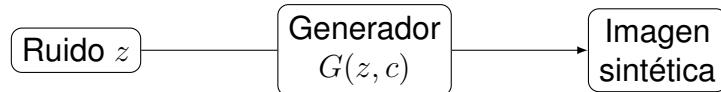


Figura 2.4: Esquema simplificado de un sistema de generación de imágenes. La condición c es opcional.

La Figura 2.4 muestra el flujo conceptual mínimo: (1) se muestrea z ; (2) el generador aplica transformaciones no lineales —habitualmente redes neuronales profundas—; (3) se obtiene la imagen sintética. Según la familia de modelos empleada (GAN, VAE o DDPM), el procedimiento de entrenamiento varía, pero el objetivo permanece: aproximar $p_{\text{data}}(x)$.

2.4.2. Aumentación de datos

La calidad y diversidad del conjunto de entrenamiento condicionan directamente el rendimiento de cualquier modelo de aprendizaje automático. Sin embargo, en dominios especializados —como la captura de UAV en vuelo— las imágenes reales suelen ser escasas, costosas o difíciles de anotar. La **aumentación de datos** ofrece una solución pragmática: generar variaciones artificiales que incrementen la cobertura de casos y reduzcan el sobreajuste [22].

2.4.3. Técnicas clásicas

Las transformaciones geométricas y fotométricas constituyen el primer nivel de aumentación:

- *Rotaciones y volteos*: simulan cambios de orientación [23].
- *Escalados y recortes*: introducen variaciones de distancia y encuadre.
- *Ajustes de color, brillo y ruido*: reproducen condiciones lumínicas y sensoriales diversas.

Técnicas automatizadas como *AutoAugment* permiten aprender políticas óptimas combinando estas operaciones [24].

2.4.4. Aumentación basada en modelos generativos

El segundo nivel de aumento se apoya en redes generativas — GAN, VAE, DDPM — capaces de producir imágenes sintéticas no presentes en el conjunto original [25]. Estas muestras amplían el dominio de variaciones posibles:

- **GAN condicionales**: generan UAV con clases, poses o atributos específicos.
- **VAE disentangled**: permiten interpolar características como envergadura o textura de forma controlada.
- **Modelos de difusión**: ofrecen fotorrealismo y control preciso sobre condiciones climáticas o de fondo [6].

2.4.5. Riesgos y buenas prácticas

Cuando la distribución sintética difiere en exceso de la real, puede introducir sesgos o artefactos indeseados. Para mitigar este riesgo, se recomienda:

- Validar la similitud con datos reales mediante métricas perceptuales, como el FID.
- Incorporar imágenes generadas en proporciones moderadas para evitar desplazar la distribución original.
- Documentar cuidadosamente licencias, procedencia y trazabilidad de cada conjunto (véase Sección 2.6).

2.4.6. Transferencia de dominio

La generación sintética de datos introduce inevitablemente una divergencia entre la distribución artificial $p_{\text{gen}}(x)$ y la distribución real $p_{\text{real}}(x)$. Esta diferencia, conocida como **brecha de dominio** o *domain gap*, puede deteriorar significativamente el rendimiento de los modelos entrenados sobre datos sintéticos cuando se despliegan en condiciones reales [26, 27]. Para mitigar este problema, surge la necesidad de aplicar técnicas de **transferencia de dominio** (*domain adaptation*), que buscan alinear ambas distribuciones con el objetivo de maximizar la generalización.

Definición general. La transferencia de dominio se refiere al conjunto de métodos diseñados para transferir conocimiento aprendido en un *dominio fuente* (por ejemplo, imágenes generadas) hacia un *dominio destino* (imágenes reales), bajo el supuesto de que ambos comparten tareas similares pero tienen distribuciones marginales distintas $p(x) \neq p'(x)$.

Implicaciones en UAVs. En el contexto de visión artificial para UAVs, este fenómeno es particularmente relevante: las imágenes sintéticas generadas pueden contener fondos simplificados, iluminación ideal o falta de occlusiones, mientras que las escenas reales presentan variabilidad ambiental, artefactos ópticos o condiciones meteorológicas que alteran sustancialmente la apariencia visual. Como resultado, los modelos discriminativos entrenados con ejemplos sintéticos tienden a sobreajustar a ese dominio y degradar su rendimiento en producción.

Estrategias principales de adaptación

Diversas técnicas han sido propuestas para reducir la brecha entre dominios. Algunas de las más relevantes incluyen:

- **Adaptación supervisada:** requiere una pequeña cantidad de muestras anotadas en el dominio real. Se basa en *fine-tuning* o *few-shot learning*, ajustando pesos de redes preentrenadas sobre sintéticos [28].
- **Adaptación no supervisada (UDA):** no requiere etiquetas en el dominio destino. Utiliza métodos adversariales (por ejemplo, *Domain-Adversarial Neural Networks* [29]) que entran un clasificador mientras un discriminador intenta distinguir entre características extraídas de ambos dominios, forzando la invariancia.
- **Transformación de estilo:** se transforman visualmente las imágenes de un dominio para parecerse al otro. Por ejemplo, se puede emplear *CycleGAN* para convertir imágenes sintéticas en realistas y viceversa, preservando la semántica [30].
- **Normalización estadística:** se modifican los histogramas de color, luminancia o distribución espectral entre dominios (por ejemplo, *Adaptive Instance Normalization*, *Histogram Matching*).

La Figura 2.5 ilustra esquemáticamente la separación entre la distribución generada $p_{\text{gen}}(x)$ y la real $p_{\text{real}}(x)$, y cómo esta afecta a la generalización del modelo.

Brecha de dominio entre datos sintéticos y reales

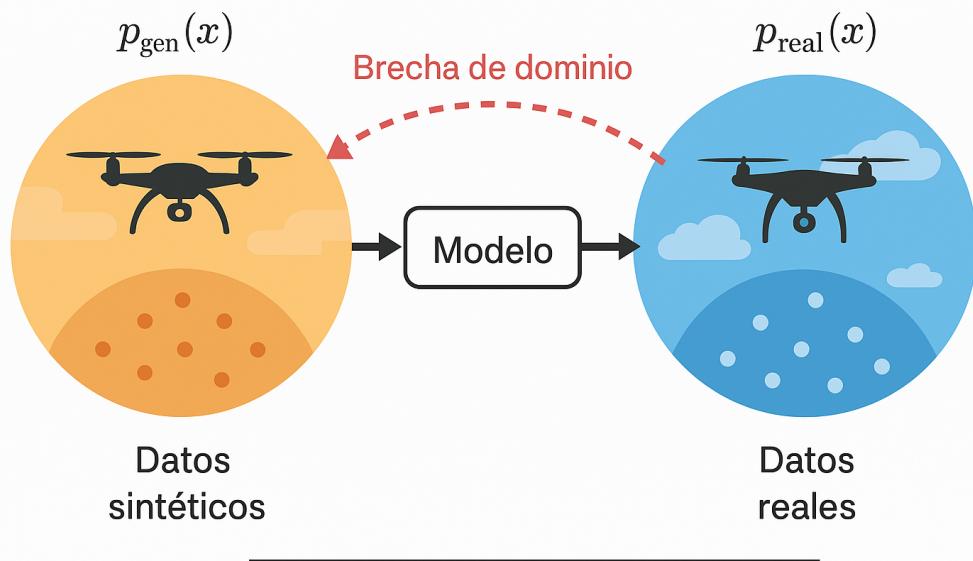


Figura 2.5: Representación esquemática de la brecha de dominio entre datos sintéticos y reales. Elaboración propia

Evaluación del impacto del dominio

Es habitual evaluar la efectividad de la transferencia mediante métricas indirectas como la caída de rendimiento entre validación en sintéticos y prueba en reales, o métricas directas como el *Maximum Mean Discrepancy* (MMD) entre espacios latentes. En el caso de UAVs, esta evaluación es crítica para asegurar que los sistemas mantienen rendimiento operativo bajo condiciones reales.

2.4.7. Sobreajuste a datos sintéticos

Aunque la generación sintética aporta una vía eficaz para aumentar la diversidad del conjunto de entrenamiento, su uso excesivo o mal calibrado puede inducir un fenómeno de sobreajuste a patrones artificiales. Este problema se manifiesta cuando el modelo aprende a explotar características específicas del conjunto sintético —como texturas poco realistas, composiciones repetitivas o fondos uniformes— que no están presentes en los datos reales [31, 32].

Este tipo de sobreajuste puede resultar en una mejora aparente de métricas como la precisión o el FID durante la fase de validación, pero conlleva una pérdida de ca-

pacidad de generalización al enfrentarse a imágenes reales no vistas. En el contexto de UAVs, esto puede implicar que un detector reconozca eficazmente drones generados con iluminación o fondos ideales, pero falle en condiciones reales con occlusiones, sombras o ruido ambiental.

Para evitar este riesgo, es fundamental:

- **Equilibrar proporciones:** mantener un ratio razonable entre datos reales y sintéticos (por ejemplo, 3:1 o 4:1).
- **Incluir variabilidad en el ruido:** usar diferentes seeds, estilos y condiciones para evitar que el modelo memorice patrones específicos del generador.
- **Validar con subconjuntos reales:** reservar una parte del conjunto de validación que no incluya imágenes generadas.
- **Evaluar transferibilidad:** probar el modelo entrenado con sintéticos sobre conjuntos completamente reales para estimar el *domain gap*.

Este fenómeno se relaciona con el conocido *domain overfitting* en aprendizaje profundo, donde el modelo se adapta demasiado al dominio de entrenamiento en detrimento del rendimiento en producción [26]. En entornos críticos como la detección de UAVs en seguridad o defensa, este riesgo adquiere una dimensión operativa clave, por lo que la generación sintética debe entenderse como una herramienta complementaria, no sustitutiva, del muestreo real.

La Tabla 2.1 resume las principales estrategias de transferencia de dominio relevantes para el tratamiento de datos generados, con énfasis en su aplicabilidad práctica al contexto UAV.

Técnica	Supervisión	Aplicación típica	Ventajas	Desventajas
Fine-tuning	Semi-supervisada	Ajuste del modelo con muestras reales	Precisión mejorada en el dominio real	Requiere anotaciones reales
DANN (adversarial)	No supervisada	Invarianza de características entre dominios	Generalización robusta entre dominios	Difícil de entrenar, inestable
CycleGAN	No supervisada	Conversión visual entre estilos (sintético → real)	Preserva semántica, reduce artefactos	No siempre preserva detalles estructurales
Normalización estadística	No supervisada	Alineamiento de distribuciones de bajo nivel	Simple de implementar, eficiente	Limitado a aspectos estadísticos simples

Tabla. 2.1: Técnicas comunes de transferencia de dominio en entornos con datos sintéticos.

Conexión con la metodología. El Capítulo 4 detalla el *pipeline* que integra ambas modalidades de aumentación y cuantifica su impacto en la capacidad de generalización de los detectores de UAV entrenados posteriormente.

2.4.8. Generación condicionada

Para los fines de este TFG resulta fundamental controlar atributos específicos: tipo de UAV, ángulo de cámara, hora del día o condiciones meteorológicas. La **generación condicionada** extiende la función G con una variable c que guía la salida [33]. Algunos enfoques destacados incluyen:

- **cGAN**: emplean etiquetas de clase para diferenciar, por ejemplo, cuadricópteros y aeronaves de ala fija.
- **Traducción de dominio (Pix2Pix, CycleGAN)**: permiten mapear modelos 3D «limpios» a fotogramas con apariencia realista.
- **Modelos texto-imagen (como Stable Diffusion)**: generan escenas completas a partir de descripciones semánticas detalladas.

El Capítulo 3 desglosa la arquitectura, función de pérdida y configuración experimental de cada uno. A su vez, el Capítulo 4 muestra cómo se integran en el *pipeline* de generación sintética con fines de entrenamiento y evaluación de sistemas de detección de objetos en vuelo.

2.5. Evaluación y métricas

Evaluar la calidad de las imágenes generadas por modelos sintéticos es un reto abierto dentro de la visión por computador y el aprendizaje profundo. Ninguna métrica actual captura de forma simultánea y fiable los tres pilares fundamentales de una buena generación: *realismo visual, diversidad semántica y fidelidad estructural*. Por ello, en la práctica se opta por una **evaluación multivista**, que combina métricas automáticas, reconstrucción comparativa y juicios humanos. Este enfoque se ha seguido en el presente trabajo, priorizando robustez experimental y compatibilidad con evaluaciones del estado del arte.

2.5.1. Métricas objetivas

Las métricas objetivas automatizadas permiten realizar comparaciones rápidas, reproducibles y escalables entre modelos. En este trabajo se han utilizado las siguientes:

FID – Frechet Inception Distance Compara estadísticamente la activación media y la covarianza de las imágenes reales y generadas usando una red Inception v3 preentrenada sobre ImageNet. Está definida como:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}} \right)$$

donde μ_r , Σ_r y μ_g , Σ_g corresponden a la media y covarianza de las representaciones para los datos reales y generados, respectivamente. Un menor valor indica una mayor similitud de distribuciones.

IS – Inception Score Se basa en la red Inception para predecir clases en las imágenes generadas. Evalúa tanto la calidad visual (cuando $p(y|x)$ es de baja entropía) como la diversidad global (alta entropía en $p(y)$). Se define como:

$$\text{IS} = \exp(\mathbb{E}_x [\text{KL}(p(y|x) \| p(y))])$$

Aunque popular, su interpretación se ve limitada cuando el conjunto de datos no coincide con las clases de entrenamiento de Inception, como sucede con UAV.

PSNR – Peak Signal-to-Noise Ratio Métrica clásica basada en píxeles, útil en tareas de reconstrucción. Mide la relación logarítmica entre la señal máxima y el error cuadrático medio (MSE) respecto a una imagen de referencia:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right)$$

A pesar de su simplicidad, no correlaciona bien con la percepción visual.

SSIM – Structural Similarity Index Propuesta como alternativa perceptual al PSNR, esta métrica modela el sistema de visión humano teniendo en cuenta luminancia, contraste y estructura:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

donde μ_x , μ_y son medias locales, σ_x , σ_y varianzas y σ_{xy} la covarianza entre bloques correspondientes.

2.5.2. Evaluación subjetiva

Aunque las métricas objetivas resultan útiles para comparar modelos, a menudo no capturan aspectos semánticos o estéticos perceptibles por humanos. Por este motivo, se han incorporado dos enfoques cualitativos:

- **Pruebas humanas de preferencia:** se ha llevado a cabo un experimento informal de evaluación A/B entre pares de imágenes generadas (StyleGAN2-ADA vs. FDM-EDM vs. WAE-MMD) para analizar qué imágenes eran preferidas por observadores no expertos, especialmente en el dominio UAV. Aunque limitada en escala, esta prueba aportó evidencia de que métricas como FID no siempre reflejan fidelidad visual percibida.
- **Observación cualitativa asistida:** se ha generado una serie de visualizaciones comparativas, organizadas horizontalmente (ver Fig. 5.19 y Fig. 5.20), para facilitar el análisis estructural, semántico y morfológico de las imágenes sintéticas frente a las reales.

2.5.3. Resumen comparativo de métricas

Las métricas presentadas en la Tabla 2.2 permiten evaluar distintos aspectos de la calidad de las imágenes generadas, pero cada una presenta sesgos y limitaciones que deben ser tenidos en cuenta al interpretar los resultados. El FID (Fréchet Inception Distance) se considera actualmente la métrica estándar para generación de imágenes, ya que estima la distancia entre distribuciones de características extraídas por una red Inception entrenada. Si bien refleja bien la fidelidad y diversidad global, puede verse afectado por la sensibilidad a valores atípicos y por la dependencia del extractor de características preentrenado.

Por su parte, el IS (Inception Score) evalúa simultáneamente la nitidez de las muestras (a través de la entropía de las predicciones) y su diversidad. Es rápido y simple de interpretar, pero tiene una cobertura limitada cuando se trabaja con dominios alejados de los datos con los que se entrenó Inception, como UAVs o imágenes técnicas.

Las métricas PSNR y SSIM son apropiadas principalmente en contextos de reconstrucción, donde existe una imagen de referencia. PSNR cuantifica errores a nivel de píxel, siendo útil como indicador de fidelidad bruta, aunque no considera aspectos perceptuales. En contraste, SSIM incorpora factores como luminancia, contraste y estructura, correlacionando mejor con la percepción humana. No obstante, su uso queda

restringido a escenarios donde existe correspondencia directa entre imágenes reales y generadas.

En conjunto, estas métricas ofrecen una evaluación complementaria: mientras que FID e IS capturan propiedades estadísticas globales y son aplicables a generación pura, SSIM y PSNR resultan más adecuados en tareas de reconstrucción. La elección de métrica debe adaptarse, por tanto, al tipo de tarea y al dominio específico en el que se trabaja.

Métrica	Evalúa	Ventajas	Limitaciones
FID	Distancia estadística entre imágenes reales y generadas (activaciones de Inception)	Captura parcialmente el realismo visual y la distribución global	Sensible a outliers y dependencia del extracto preentrenado
IS	Claridad individual + diversidad global (Inception)	Rápido de calcular y fácil de interpretar	Insuficiente para dominios fuera de ImageNet
SSIM	Similitud perceptual local (estructura + contraste)	Alineado con percepción humana en reconstrucción	Inaplicable a generación sin referencia
PSNR	Fidelidad de píxel a píxel respecto a una referencia	Intuitivo y ampliamente conocido	Métrica puramente numérica, sin criterio perceptual

Tabla. 2.2: Comparativa de métricas utilizadas para evaluar la calidad generativa.

2.5.4. Criterio adoptado en este trabajo

El protocolo de evaluación empleado en este TFG ha seguido una estrategia mixta, con el objetivo de maximizar la objetividad sin perder alineamiento con la percepción visual. Concretamente:

1. **FID e IS** han sido aplicadas sistemáticamente a todos los modelos generativos en ambas tareas (reconstrucción y síntesis libre), sobre los conjuntos CelebA y UAV.
2. **PSNR y SSIM** se han reservado para los modelos de reconstrucción VAE, donde existe una imagen original como referencia directa.
3. **Evaluación cualitativa** mediante visualizaciones y pruebas humanas ha servido como mecanismo de validación cruzada para interpretar mejor los resultados métricos y detectar errores perceptuales no capturados por los indicadores numéricos.

Esta combinación ha permitido establecer una base sólida de análisis, coherente con los criterios de la literatura actual y sensible a las particularidades del dominio UAV.

2.6. Consideraciones éticas y de uso responsable

El desarrollo de modelos generativos con capacidad para producir imágenes casi indistinguibles de la realidad amplía tanto el espectro de aplicaciones como los posibles riesgos. A continuación se abordan los vectores de impacto más relevantes, contextualizados en la síntesis de UAV, y se esbozan prácticas de mitigación que orientan la ejecución experimental de este TFG.

2.6.1. Posibles riesgos

1. **Desinformación y *deepfakes*:** las imágenes manipuladas pueden utilizarse con fines de propaganda, sabotaje o manipulación. En el ámbito militar, fotogramas falsificados de UAV podrían generar falsas alarmas o encubrir incursiones reales [34].
2. **Sesgo en los datos:** los modelos generativos tienden a replicar los desequilibrios presentes en su conjunto de entrenamiento. La infrarepresentación de drones pequeños o de ciertos tipos de entornos puede degradar el rendimiento de los sistemas en condiciones reales [27].
3. **Privacidad y derechos de autor:** el uso de imágenes obtenidas mediante scraping puede infringir licencias o vulnerar normativas de protección de datos personales, como el Reglamento General de Protección de Datos (RGPD) [35].
4. **Huella de carbono:** el entrenamiento de modelos generativos de gran escala implica un consumo energético considerable, con impacto ambiental directo [36].
5. **Uso dual:** la misma tecnología que refuerza capacidades defensivas —como la vigilancia o la detección de amenazas— puede utilizarse para perfeccionar sistemas ofensivos. En este sentido, el borrador del *AI Act* europeo propone clasificar estos sistemas como de “alto riesgo” [37].

2.7. Conexión con el proyecto

Los fundamentos expuestos en este capítulo —relativos a la síntesis de imágenes, la aumentación de datos y la transferencia de dominio— tienen una aplicación directa y estratégica en la problemática abordada por este Trabajo de Fin de Grado: la generación automática de imágenes de vehículos aéreos no tripulados (UAVs) con fines de entrenamiento y validación de sistemas de visión artificial.

Limitaciones del conjunto de datos real

En contextos reales, la adquisición de datos de UAVs con calidad suficiente para entrenar modelos de detección presenta varias limitaciones:

- **Escasez de ejemplos anotados:** especialmente en condiciones extremas (baja iluminación, occlusiones parciales, distancias elevadas).
- **Coste operativo elevado:** requiere despliegue físico, planificación de vuelo y personal autorizado.
- **Sesgo de representatividad:** los conjuntos reales tienden a concentrarse en ciertos modelos de dron o en determinados entornos (urbanos, cielos despejados), lo que puede afectar la generalización.

Estas limitaciones hacen que el entrenamiento basado exclusivamente en datos reales esté expuesto a una alta varianza y riesgo de *underfitting* o sobreajuste a situaciones específicas.

Generación sintética como solución complementaria

La generación automática de imágenes sintéticas mediante modelos como GAN, VAE o DDPM permite:

- **Incrementar la diversidad visual:** controlando explícitamente variables como la perspectiva, el fondo, la meteorología o el tipo de UAV.
- **Reproducir condiciones raras:** que son difíciles de capturar en escenarios reales, como contraluces, sombras, niebla o posiciones extremas de cámara.

- **Reducir la necesidad de anotación manual:** al incorporar metadatos directamente desde el generador, especialmente en generación condicionada.

Esta capacidad de expansión controlada del dataset permite mejorar la cobertura del espacio de entrada sin necesidad de aumentar exponencialmente el número de muestras reales capturadas.

Integración con la aumentación tradicional y supervisada

La generación de imágenes no sustituye por completo la aumentación tradicional, sino que se complementa con ella. En el pipeline propuesto, se contemplan dos niveles de aumentación:

- **Transformaciones geométricas y fotométricas** (rotaciones, escalados, distorsiones, cambios de brillo o contraste), aplicadas sobre datos reales y sintéticos.
- **Generación condicional de muestras** con modelos como cGAN o Diffusion, que permiten extender regiones del espacio de atributos poco cubiertas por los datos originales.

Este enfoque híbrido proporciona una base más sólida para el entrenamiento de detectores robustos, al tiempo que preserva la trazabilidad y control sobre la procedencia de los datos.

Síntesis

El flujo metodológico del proyecto parte de dos fuentes de datos: conjuntos reales (como CelebA y UAV Detection Dataset) e imágenes sintéticas generadas mediante arquitecturas como GAN, VAE o modelos de difusión. Ambos tipos de datos se someten a procesos de aumentación, que incluyen desde técnicas clásicas (rotaciones, escalados, cambios de brillo o ruido) hasta métodos más avanzados basados en interpolaciones latentes o generación condicional.

Una vez preprocesados —con operaciones como redimensionado a 64×64 , normalización y limpieza—, los datos alimentan el entrenamiento de modelos discriminativos o reconstructivos. Estos modelos son evaluados mediante un conjunto de métricas

cuantitativas (FID, IS, PSNR, SSIM) que permiten valorar tanto la fidelidad visual como la capacidad de generalización.

Este flujo modular e iterativo garantiza que cada componente (generación, aumentación, entrenamiento, evaluación) contribuya de forma controlada y medible al objetivo final del trabajo: mejorar el análisis visual de UAVs mediante datos sintéticos robustos y controlables.

En conjunto, este enfoque permite enfrentar los desafíos propios del dominio UAV con una base metodológica sólida, escalable y reproducible, tal como se desarrolla en los siguientes capítulos del trabajo.

Capítulo 3

Modelos Generativos

3.1. Introducción

En apenas una década, los modelos generativos han pasado de ser un artefacto de laboratorio a convertirse en el motor de buena parte de la innovación en visión por computador, síntesis de datos y creación de contenido. La Figura 3.1 resume los hitos más representativos: desde la aparición de las *Redes Generativas Adversariales* (GAN) en 2014 [3], los *Autoencoders Variacionales* (VAE) en 2013–2014 [4, 5], hasta la irrupción de los *Modelos de Difusión* y sus variantes latentes entre 2020 y 2024 [6, 7, 8]. Cada avance ha venido acompañado de mejoras en estabilidad, calidad y control, así como de nuevas aplicaciones que abarcan desde el diseño asistido hasta la medicina.

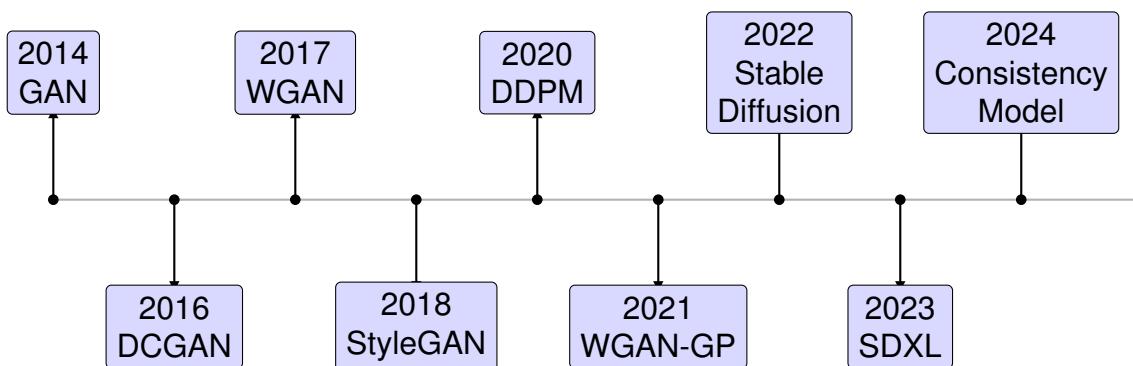


Figura 3.1: Línea temporal de los hitos clave en modelos generativos profundos (2014–2024). Elaboración propia

El presente trabajo se sitúa en el contexto de la **detección y seguimiento de vehículos aéreos no tripulados (UAVs)**. En estos escenarios, la obtención de datos reales resulta costosa y, a menudo, limitada por condiciones de vuelo, regulaciones

y cuestiones de privacidad. Disponer de grandes cantidades de *imágenes sintéticas realistas* —donde el dron aparece en distintas alturas, ángulos y fondos— se vuelve crucial para entrenar detectores robustos. Asimismo, la capacidad de controlar atributos como la iluminación, la altitud o el tipo de dron facilita la generación de *casos de borde* que, de otro modo, serían difíciles de capturar in situ.

Con este objetivo, el capítulo revisa de forma estructurada las tres familias que han marcado la evolución reciente del aprendizaje generativo:

- **Autoencoders Variacionales (VAE)**: primeros métodos probabilísticos que aportan un espacio latente interpretable, útiles para tareas de interpolación y edición.
- **Redes Generativas Adversariales (GAN)**: modelos basados en juegos competitivos, capaces de producir imágenes de alta fidelidad con tiempos de inferencia mínimos.
- **Modelos de Difusión (DDPM)**: enfoques inspirados en procesos estocásticos, que —a costa de mayor cómputo— logran el estado del arte en calidad visual y control semántico.

Durante el recorrido se destacarán los avances clave que han permitido superar problemas clásicos —como el *mode collapse* en GAN [38], la borrosidad en VAE [39] o el elevado número de pasos en DDPM [40]— y se ilustrará cómo cada técnica se adapta (o no) a la síntesis de escenas con UAVs.

Por último, se señalarán brevemente las implicaciones éticas del contenido generado (sesgo, usos maliciosos), remitiendo al análisis detallado del Capítulo 2.6. De este modo, el lector dispondrá del marco conceptual necesario antes de adentrarse en los experimentos y resultados que vertebran el resto de este Trabajo Fin de Grado.

3.2. Modelos generativos: panorámica

Los avances recientes en generación de imágenes se articulan, en gran medida, alrededor de tres familias de modelos profundos. Cada una aborda el problema desde una perspectiva distinta: juego adversarial, inferencia variacional o difusión estocástica. El objetivo de esta sección es presentar sus conceptos troncales y justificar por qué constituyen los pilares del Capítulo 3. El desarrollo matemático completo se delega a dicho capítulo.

3.2.1. Evolución histórica

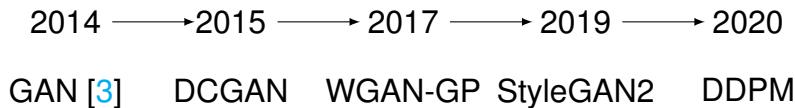


Figura 3.2: Hitos abreviados en la evolución de los modelos generativos profundos.

3.2.2. Tres enfoques predominantes

GAN Introducen una dinámica adversarial entre generador y discriminador para aproximar p_{data} . Se caracterizan por su velocidad de inferencia y alta nitidez visual, aunque sufren inestabilidad durante el entrenamiento y fenómenos como el *mode collapse*. Variantes destacadas: DCGAN, WGAN-GP, StyleGAN2.

VAE Basados en inferencia variacional, modelan explícitamente $p(z|x)$ y un prior $p(z)$. Proporcionan un espacio latente interpretable y disentangled, aunque tienden a generar imágenes de menor nitidez. Variantes comunes: β -VAE, VQ-VAE, WAE.

DDPM Reformulan la generación como la inversión de un proceso difusivo de ruido. Alcanzan el mayor grado de fotorrealismo y estabilidad, a costa de tiempos de inferencia elevados. Principales evoluciones: Improved DDPM, Latent Diffusion, Stable Diffusion.

3.2.3. Comparación cualitativa

Criterio	GAN	VAE	DDPM
Calidad visual	Alta	Media	Muy alta
Estabilidad de entrenamiento	Baja	Alta	Muy alta
Velocidad de inferencia	~ ms	~ ms	> 1 s (multi-step)
Latente interpretable	Bajo	Alto	Medio
Uso típico	Arte, super-resolución	Disentanglement, compresión	Texto-imagen, fotos hiperrealistas

Tabla. 3.1: Visión de alto nivel de las fortalezas y debilidades de cada familia.

3.3. Autoencoders Variacionales (VAE)

3.3.1. Fundamentos teóricos

Los Autoencoders Variacionales (VAE), introducidos por Kingma y Welling [4] y Rezende et al. [5], reformulan el autoencoder tradicional como un modelo generativo probabilístico. En lugar de aprender una codificación determinista, el VAE modela una distribución latente $q_\phi(z|x)$ desde la que se puede muestrear para reconstruir los datos originales.

Objetivo de aprendizaje

El objetivo principal es maximizar la evidencia marginal del dato observado:

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)} \right] + \text{KL}(q_\phi(z|x) \| p_\theta(z|x)) \quad (3.1)$$

Como $p_\theta(z|x)$ es inalcanzable directamente, se maximiza la evidencia inferior (ELBO):

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] - \text{KL}(q_\phi(z|x) \| p(z)) \quad (3.2)$$

- El primer término representa la calidad de reconstrucción.
- El segundo término actúa como regularizador, forzando $q_\phi(z|x)$ a acercarse al prior $p(z) = \mathcal{N}(0, I)$.

Truco de reparametrización

Para poder realizar la retropropagación a través del muestreo estocástico en los Autoencoders Variacionales (VAE), se emplea el llamado **truco de reparametrización**. Este permite convertir una operación de muestreo no diferenciable en una expresión diferenciable, facilitando así el entrenamiento del modelo mediante gradientes.

La clave consiste en expresar la variable latente z como una transformación determinista de una variable aleatoria auxiliar $\varepsilon \sim \mathcal{N}(0, I)$, separando la fuente de aleatoriedad del resto del modelo. En lugar de muestrear directamente $z \sim \mathcal{N}(\mu, \sigma^2)$, se utiliza la siguiente expresión:

$$z = \mu(x) + \sigma(x) \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I) \quad (3.3)$$

Aquí, tanto $\mu(x)$ como $\sigma(x)$ son funciones aprendidas por el encoder, y la variable ε introduce la estocasticidad. Gracias a esta reformulación, el proceso completo se mantiene diferenciable, permitiendo optimizar la función de pérdida mediante retropropagación.

Este mecanismo es esencial en los VAE, ya que permite ajustar conjuntamente la red generadora y la distribución aproximada $q_\phi(z | x)$, dentro del marco de la inferencia variacional.

3.3.2. Arquitectura genérica

La Figura 3.3 ilustra la arquitectura genérica de un *Autoencoder Variacional* (VAE). Está compuesta por un codificador $q_\phi(z | x)$, una etapa de reparametrización que permite el muestreo del espacio latente mediante $z = \mu + \sigma \cdot \varepsilon$, y un decodificador $p_\theta(x | z)$ encargado de reconstruir la entrada original x . Esta configuración permite la optimización mediante descenso de gradiente, incluso con la introducción de ruido estocástico.

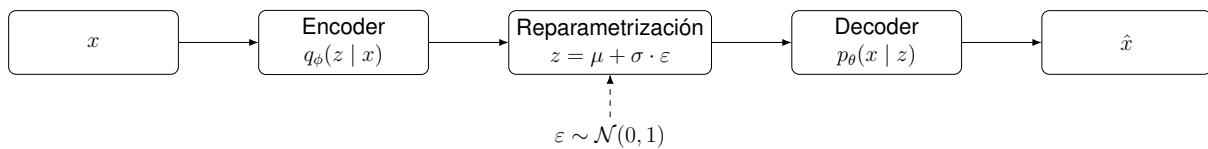


Figura 3.3: Esquema general de un Autoencoder Variacional (VAE), con codificación, muestreo y decodificación. Elaboración propia

3.3.3. Implementación básica en PyTorch

```

1  class VAE(nn.Module):
2      def __init__(self, latent_dim):
3          super().__init__()
4          self.encoder = nn.Sequential(nn.Linear(784, 400), nn.ReLU())
5          self.mu = nn.Linear(400, latent_dim)
6          self.logvar = nn.Linear(400, latent_dim)
7          self.decoder = nn.Sequential(
8              nn.Linear(latent_dim, 400), nn.ReLU(),
9              nn.Linear(400, 784), nn.Sigmoid()
10         )
11
12     def reparameterize(self, mu, logvar):
13         std = torch.exp(0.5 * logvar)
14         eps = torch.randn_like(std)
15         return mu + eps * std
16
17     def forward(self, x):
18         h = self.encoder(x)
19         mu, logvar = self.mu(h), self.logvar(h)
20         z = self.reparameterize(mu, logvar)
21         return self.decoder(z), mu, logvar

```

Listado 3.1: Implementación básica de un VAE en PyTorch

3.3.4. Aplicaciones a UAVs

- **Interpolación latente:** permite generar transiciones suaves entre distintos tipos de drones.
- **Reconstrucción estructurada:** ideal para recuperar frames degradados desde representaciones comprimidas.
- **Compresión semántica:** NVAE y derivados logran tasas superiores a JPEG en escenas con fondos simples como el cielo.

3.3.5. Principales variantes del VAE

A lo largo de los últimos años se han propuesto múltiples variantes del modelo VAE original, diseñadas para mejorar la disentanglement, la fidelidad visual o la eficiencia de la reconstrucción. En las siguientes subsecciones se presentan algunas de las más relevantes.

β -VAE

El modelo β -VAE fue propuesto por Higgins et al. [41] para fomentar la disentanglement de factores latentes. Introduce un hiperparámetro $\beta > 1$ en la ELBO, escalando la penalización KL:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta \cdot \text{KL}(q_\phi(z|x) \| p(z)) \quad (3.4)$$

Factores disentangled

Se habla de representación *disentangled* cuando cada dimensión del espacio latente z controla un atributo semántico independiente en la imagen generada. Por ejemplo, un componente de z podría controlar únicamente la orientación del dron, mientras otro regula su tamaño o el color del fondo. Esta separación favorece el control interpretable de la generación. **Ejemplos de atributos disentangled útiles en UAV:**

- Tamaño del UAV.
- Tipo de propulsión.
- Altitud o posición angular.

Ventajas:

- Facilita el control sobre atributos latentes.
- Aporta interpretabilidad al espacio z .

Limitaciones:

- Altos valores de β pueden deteriorar la calidad de reconstrucción.

InfoVAE

El modelo InfoVAE, propuesto por Zhao et al. [42], surge como una respuesta al desequilibrio que se produce en el entrenamiento estándar de los VAE entre la fidelidad de reconstrucción y la regularización del espacio latente.

A diferencia del β -VAE, que penaliza más fuerte la divergencia KL, InfoVAE introduce una nueva formulación de la función de pérdida que permite maximizar explícitamente la información mutua entre las variables latentes z y los datos observables x .

Función objetivo InfoVAE reemplaza la ELBO clásica por una variante con control flexible sobre la regularización:

$$\mathcal{L}_{\text{InfoVAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \alpha \cdot \text{KL}(q_\phi(z|x) \| p(z)) + \lambda \cdot D(q(z), p(z)) \quad (3.5)$$

Donde:

- $\alpha \in [0, 1]$ regula el peso de la divergencia condicional.
- λ permite forzar la alineación entre las distribuciones agregadas.
- D puede ser MMD, JS o cualquier divergencia entre distribuciones marginales.

Ventajas:

- Mayor control sobre la información preservada en el espacio latente.
- Capacidad para evitar el *posterior collapse*, donde $q(z|x) \approx p(z)$.
- Aplicable a tareas donde es crucial preservar la estructura semántica del dato original.

WAE-MMD

Los Wasserstein Autoencoders (WAE), introducidos por Tolstikhin et al. [43], proponen una reinterpretación de los VAE basada en el principio de optimal transport. En lugar de minimizar directamente la divergencia KL entre $q_\phi(z|x)$ y el prior $p(z)$, el WAE minimiza la distancia entre las distribuciones marginales agregadas $q(z) = \int q_\phi(z|x)p(x)dx$ y $p(z)$.

Función objetivo:

$$\mathcal{L}_{\text{WAE}} = \mathbb{E}_{p(x)} \mathbb{E}_{q_\phi(z|x)}[c(x, G(z))] + \lambda \cdot D(q(z), p(z)) \quad (3.6)$$

Donde $c(x, G(z))$ es un coste de reconstrucción (usualmente MSE) y D es una divergencia entre las distribuciones latentes, que puede ser la MMD (Maximum Mean Discrepancy), resultando en la variante WAE-MMD.

Ventajas:

- Mayor flexibilidad al elegir la métrica entre distribuciones.
- Mejor desempeño en tareas donde se requiere una codificación estructurada y continua.
- Compatible con regularización suave que evita el colapso del latente.

VQ-VAE

Los Vector Quantised-Variational Autoencoders (VQ-VAE), propuestos por van den Oord et al. [44], sustituyen el espacio latente continuo de los VAE por un espacio discreto representado mediante un diccionario de vectores latentes aprendibles.

Fucionamiento En lugar de muestrear desde una distribución continua, el encoder produce una representación $z_e(x)$ que se cuantiza al vector más cercano de una tabla de códigos $\{e_k\}_{k=1}^K$. El decoder genera la reconstrucción a partir del código discretizado:

$$q(z = e_k|x) = \begin{cases} 1, & \text{si } k = \arg \min_j \|z_e(x) - e_j\|_2 \\ 0, & \text{en otro caso} \end{cases} \quad (3.7)$$

Función de pérdida La pérdida se compone de tres términos:

- Reconstrucción: $\|x - \hat{x}\|^2$
- Commitment loss: fuerza al encoder a comprometerse con una entrada cercana al código e_k
- Codebook loss: actualiza el diccionario de vectores $\{e_k\}$

Ventajas:

- Facilita el uso de modelos generativos secuenciales como Transformers.
- Permite codificar con mayor compresión sin perder semántica.
- Estabilidad en el entrenamiento al evitar muestreo continuo.

NVAE

El modelo NVAE (Neural Variational Autoencoder), propuesto por Vahdat y Kautz [39], representa una de las aproximaciones más avanzadas en la arquitectura de auto-encoders variacionales jerárquicos. Su principal innovación es el uso de múltiples niveles de variables latentes organizados jerárquicamente junto con bloques convolucionales residuales profundos.

Estructura jerárquica NVAE implementa una pila de capas latentes organizadas de arriba hacia abajo, permitiendo que la información fluya desde latentes de alto nivel (semántica global) hasta latentes de bajo nivel (detalles locales):

- Cada nivel latente z_i se modela condicionalmente dado los anteriores $z_{>i}$.
- Se utiliza una arquitectura similar a U-Net con skip connections.

Función objetivo Se sigue maximizando la ELBO, pero con una descomposición jerárquica de la divergencia KL:

$$\mathcal{L}_{\text{NVAE}} = \mathbb{E}_{q(z_1, \dots, z_L|x)} [\log p(x|z_1, \dots, z_L)] - \sum_{i=1}^L \text{KL}(q(z_i|x) \| p(z_i|z_{>i})) \quad (3.8)$$

Ventajas:

- Mejora la capacidad generativa en resoluciones altas.
- Captura representaciones latentes más ricas y profundas.
- Competitivo con GANs y modelos de difusión en datasets complejos.

Diffusion-VAE

La variante Diffusion-VAE [45] combina la estructura probabilística de los autoencoders variacionales con la capacidad generativa de los modelos de difusión. En este enfoque, el decodificador $p_\theta(x|z)$ se reemplaza por un modelo de difusión condicional, lo que permite mejorar la calidad visual sin perder interpretabilidad latente.

Arquitectura híbrida

- El encoder sigue siendo una red neuronal que estima $q_\phi(z|x)$.
- El decoder es un modelo de difusión condicional $p_\theta(x_t|x_{t+1}, z)$, entrenado para denoising a partir del código latente z .

Función objetivo Se modifica la ELBO tradicional incorporando la pérdida de denoising del modelo de difusión:

$$\mathcal{L}_{\text{Diffusion-VAE}} = \mathbb{E}_{q_\phi(z|x)} \left[\sum_{t=1}^T \mathbb{E}_{q(x_t|x)} [\|x_{t-1} - D_\theta(x_t, z, t)\|^2] \right] + \beta \cdot \text{KL}(q_\phi(z|x) \| p(z)) \quad (3.9)$$

Ventajas:

- Generación de imágenes de alta calidad sin artefactos típicos de VAE o GAN.
- Permite mantener el espacio latente interpretable y navegable.
- Mejora la diversidad visual en conjuntos pequeños o ruidosos.

3.3.6. Valoración del enfoque VAE

Las variantes del modelo VAE analizadas en esta sección muestran una clara evolución desde enfoques básicos orientados a la regularización latente (β -VAE), hasta arquitecturas jerárquicas profundas (NVAE) y métodos híbridos con difusión (Diffusion-VAE). Cada una ofrece ventajas particulares:

- **Interpretabilidad y control semántico:** β -VAE, InfoVAE.
- **Fidelidad en la reconstrucción:** WAE-MMD, VQ-VAE.
- **Capacidad generativa en alta resolución:** NVAE, Diffusion-VAE.

En el contexto UAV, los resultados experimentales muestran que WAE-MMD y Diffusion-VAE son especialmente eficaces cuando se requiere alta precisión visual, mientras que InfoVAE y VQ-VAE resultan útiles para tareas de codificación semántica y control de atributos.

Este análisis proporciona una base sólida para comparar los VAE con otras familias generativas como GANs y modelos de difusión, que se abordarán en los próximos apartados.

3.4. Redes Generativas Adversariales (GAN)

Las Redes Generativas Adversariales (GAN), introducidas por Goodfellow et al. [3], constituyen una de las arquitecturas más influyentes en la generación de imágenes sintéticas. Su principal innovación radica en el enfoque adversarial, en el que dos redes neuronales —un generador G y un discriminador D — compiten en un juego de suma cero.

3.4.1. Fundamentos teóricos

El generador $G(z)$ aprende a mapear ruido aleatorio $z \sim \mathcal{N}(0, I)$ al espacio de datos x , mientras que el discriminador $D(x)$ intenta distinguir entre muestras reales y sintéticas. La función de pérdida original se plantea como un problema minimax:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] \quad (3.10)$$

La solución óptima ocurre cuando $p_G(x) = p_{\text{data}}(x)$, es decir, cuando el generador produce muestras indistinguibles de los datos reales.

Problemas típicos A pesar de su potencial, las GANs presentan retos técnicos relevantes:

- **Colapso de modo:** el generador produce un subconjunto limitado de muestras.
- **Oscilaciones en el entrenamiento:** la competencia entre G y D puede dificultar la convergencia.
- **Gradientes débiles:** en etapas iniciales, D puede dominar y generar gradientes no informativos.

Soluciones y variantes Numerosas variantes han abordado estos problemas, incluyendo:

- **WGAN:** reformulación con distancia de Wasserstein.
- **WGAN-GP:** penalización del gradiente para imponer la condición 1-Lipschitz.
- **StyleGAN:** arquitectura basada en estilos para control detallado.

3.4.2. Arquitectura genérica

La figura 3.4 muestra la arquitectura genérica de una GAN: el generador produce muestras sintéticas x_g que el discriminador intenta diferenciar de las reales $x \sim p_{\text{data}}$. La salida del discriminador es una probabilidad $D(x) \in [0, 1]$

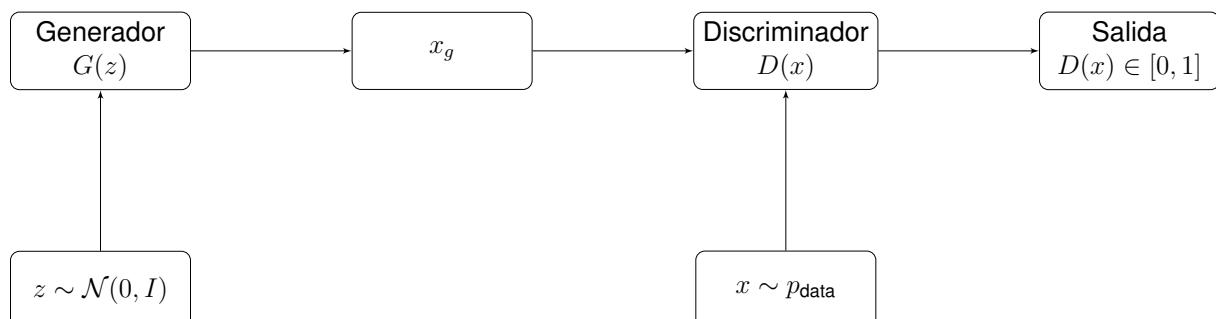


Figura 3.4: Esquema básico de una GAN. Elaboración propia

3.4.3. Implementación básica en PyTorch

```
1 class Generator(nn.Module):
2     def __init__(self):
3         super().__init__()
4         self.net = nn.Sequential(
5             nn.Linear(100, 128), nn.ReLU(),
6             nn.Linear(128, 784), nn.Tanh()
7         )
8
9     def forward(self, z):
10        return self.net(z)
11
12 class Discriminator(nn.Module):
13     def __init__(self):
14         super().__init__()
15         self.net = nn.Sequential(
16             nn.Linear(784, 128), nn.LeakyReLU(0.2),
17             nn.Linear(128, 1), nn.Sigmoid()
18         )
19
20     def forward(self, x):
21        return self.net(x)
```

Listado 3.2: Implementación básica de una GAN en PyTorch

3.4.4. Aplicaciones a UAVs

Las GANs se han empleado con éxito en el contexto de vehículos aéreos no tripulados (UAV) para mejorar la diversidad y robustez de los sistemas de visión artificial. Sus principales aplicaciones incluyen:

- **Síntesis de escenarios adversos:** generación de UAVs bajo condiciones poco frecuentes (iluminación extrema, rotación atípica, fondos urbanos complejos) para robustecer detectores discriminativos como YOLO o SSD.
- **Dominio cruzado:** uso de modelos como CycleGAN para traducir imágenes simuladas a estilo realista, reduciendo el dominio gap entre datasets sintéticos y reales.
- **Superresolución:** GANs como ESRGAN permiten mejorar la calidad de capturas con baja resolución o alta distancia, lo que es útil en aplicaciones con UAVs de vigilancia.
- **Aumentación de datos:** generación controlada de nuevas muestras UAV a partir de atributos como tipo de dron, orientación o altitud.

Estas capacidades convierten a las GAN en herramientas clave para ampliar y diversificar conjuntos de datos UAV, especialmente cuando el número de imágenes reales es limitado.

3.4.5. Principales variantes

En esta sección se describen algunas de las variantes más relevantes de las GANs, que han surgido para abordar limitaciones del modelo original. Estas variantes introducen mejoras en la arquitectura, la función de pérdida o los métodos de regularización, y han demostrado ser especialmente útiles en tareas de generación de imágenes como las que se presentan en este trabajo.

DCGAN

La arquitectura Deep Convolutional GAN (DCGAN), propuesta por Radford et al. [46], introduce una serie de buenas prácticas estructurales que mejoran significativamente la estabilidad y calidad de las GANs básicas.

Características principales:

- Sustituye capas lineales por convoluciones (transpuestas en el generador).
- Elimina el uso de capas de pooling, reemplazándolas por strides.
- Aplica Batch Normalization en ambas redes.
- Usa ReLU en el generador (excepto salida: tanh) y LeakyReLU en el discriminador.

Arquitectura típica:

- Entrada del generador: vector latente $z \sim \mathcal{N}(0, I)$ de 100 dimensiones.
- Salida del generador: imagen RGB de 64x64 píxeles.
- Discriminador: convoluciones descendentes hasta un único valor escalar $D(x)$.

Ventajas:

- Entrenamiento más estable que GANs con capas densas.
- Mejora en la coherencia visual de las muestras generadas.
- Arquitectura fácilmente escalable a otras resoluciones.

WGAN

La variante Wasserstein GAN (WGAN), propuesta por Arjovsky et al. [47], reformula la función objetivo de las GAN para mejorar la estabilidad del entrenamiento y reducir el colapso de modo. En lugar de la divergencia de Jensen-Shannon utilizada en la GAN original, WGAN minimiza la distancia de Wasserstein (también conocida como Earth Mover's Distance).

Función objetivo:

$$\min_G \max_{D \in \mathcal{D}_1} \mathbb{E}_{x \sim p_{\text{data}}}[D(x)] - \mathbb{E}_{z \sim p(z)}[D(G(z))] \quad (3.11)$$

Donde \mathcal{D}_1 es el conjunto de funciones 1-Lipschitz.

Implementación práctica: Para asegurar la condición de 1-Lipschitz, los autores proponen aplicar *weight clipping* a los parámetros del discriminador.

Ventajas:

- Métrica continua y diferenciable que correlaciona mejor con la calidad visual.
- Reducción significativa de la inestabilidad y oscilaciones en el entrenamiento.
- Facilita el análisis de la convergencia y diagnóstico del aprendizaje.

Limitaciones:

- El clipping excesivo puede limitar la capacidad expresiva del discriminador.
- El tuning del rango de clipping requiere validación empírica.

WGAN-GP

La variante WGAN-GP (Wasserstein GAN with Gradient Penalty), propuesta por Gulrajani et al. [48], mejora la formulación de WGAN sustituyendo el *weight clipping* por una penalización suave sobre el gradiente del discriminador. Esto permite imponer de forma más efectiva la condición de 1-Lipschitz, mejorando la estabilidad y expresividad del modelo.

Función objetivo:

$$\mathcal{L}_{\text{WGAN-GP}} = \mathbb{E}_{D(G(z))} - \mathbb{E}_{D(x)} + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (3.12)$$

- \hat{x} son interpolaciones entre muestras reales y generadas.
- λ controla la intensidad de la penalización (usualmente $\lambda = 10$).

Ventajas:

- Elimina la necesidad de clipping, preservando la capacidad representativa del discriminador.
- Mejora la estabilidad del entrenamiento incluso en arquitecturas profundas.
- Reduce significativamente el colapso de modo frente a GAN clásicas.

LSGAN

Least Squares GAN (LSGAN), propuesto por Mao et al. [49], modifica la función de pérdida adversarial para utilizar el error cuadrático medio (MSE) en lugar del logaritmo de probabilidad. Esto suaviza las actualizaciones de gradiente y ayuda a evitar saturaciones, haciendo el entrenamiento más estable. **Función objetivo:**

Para el discriminador:

$$\mathcal{L}_D = \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p(z)} [(D(G(z)) - a)^2] \quad (3.13)$$

Para el generador:

$$\mathcal{L}_G = \frac{1}{2} \mathbb{E}_{z \sim p(z)} [(D(G(z)) - c)^2] \quad (3.14)$$

Donde los valores típicos son $a = 0$, $b = 1$ y $c = 1$.

Ventajas:

- Genera gradientes más informativos cuando el discriminador es confiado.
- Mejora la cobertura de modos frente a GANs con pérdida binaria.
- Fácil de implementar y compatible con arquitecturas estándar.

RaGAN

Relativistic Average GAN (RaGAN), propuesto por Jolicoeur-Martineau [50], introduce un cambio conceptual en la función del discriminador: en lugar de predecir la probabilidad absoluta de que una muestra sea real, predice la probabilidad de que una muestra sea más real que otra en promedio. Esto busca aumentar la competencia entre el generador y el discriminador desde una perspectiva más comparativa.

Función objetivo: La pérdida del discriminador se define como:

$$\mathcal{L}_D = -\mathbb{E}_{x_r}[\log(\text{sigmoid}(D(x_r) - \mathbb{E}_{x_f} D(x_f)))] - \mathbb{E}_{x_f}[\log(\text{sigmoid}(D(x_f) - \mathbb{E}_{x_r} D(x_r)))] \quad (3.15)$$

Y la del generador como:

$$\mathcal{L}_G = -\mathbb{E}_{x_r}[\log(\text{sigmoid}(D(x_f) - \mathbb{E}_{x_r} D(x_r)))] - \mathbb{E}_{x_f}[\log(\text{sigmoid}(D(x_r) - \mathbb{E}_{x_f} D(x_f)))] \quad (3.16)$$

Ventajas:

- Mejora la estabilidad del entrenamiento sin requerir modificaciones arquitectónicas.
- Aumenta la sensibilidad a diferencias sutiles entre muestras reales y falsas.
- Compatible con cualquier arquitectura de GAN existente.

CycleGAN

CycleGAN, propuesto por Zhu et al. [30], permite traducir imágenes entre dos dominios sin necesidad de pares de entrenamiento alineados. Esto es especialmente útil cuando no existen datasets emparejados entre los estilos de origen y destino (por ejemplo, UAV simulados versus UAV reales).

Funcionamiento: CycleGAN introduce dos generadores $G : X \rightarrow Y$ y $F : Y \rightarrow X$, así como dos discriminadores D_Y y D_X , entrenados conjuntamente con una pérdida de consistencia cíclica:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim X}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim Y}[\|G(F(y)) - y\|_1] \quad (3.17)$$

Ventajas:

- No requiere pares alineados, lo que reduce el coste de anotación.
- Preserva la estructura semántica de las imágenes durante la traducción.
- Ampliamente adoptado en tareas de transferencia de estilo realista.

StyleGAN

StyleGAN, introducido por Karras et al. [51], representa un avance significativo en la calidad y control de la generación de imágenes sintéticas. Su arquitectura introduce un mapeo separado entre el espacio latente z y un nuevo espacio intermedio w , desde el cual se inyecta información estilo a cada nivel de resolución del generador.

Innovaciones clave:

- **Espacio latente intermedio w :** permite una manipulación semántica más interpretable.
- **Modulación de estilo:** cada bloque convolucional recibe escalamiento y sesgo específicos derivados de w .
- **Ruido estocástico:** mejora la variación de detalles locales como texturas.
- **Progresión de resolución:** entrenamiento que comienza desde imágenes de baja resolución y se incrementa progresivamente.

Mejoras en StyleGAN2: StyleGAN2 [52] introduce cambios importantes para eliminar artefactos visibles en los resultados generados:

- Sustitución de normalización por pixel-norm con demodulación de pesos.
- Uso de convoluciones más profundas y filtro FIR para aliasing.
- Arquitectura más limpia que permite entrenamientos más estables.

Entrenamiento con ADA: StyleGAN2-ADA [53] propone el uso de *Adaptive Discriminator Augmentation*, un mecanismo que aplica aumentos de datos de forma adaptativa únicamente cuando el discriminador empieza a sobreajustar. Esto es crucial en escenarios con pocos datos.

Ventajas:

- Control preciso sobre atributos faciales, morfológicos o de estilo.
- Generación de imágenes fotorrealistas de alta resolución (1024x1024+).
- Capacidad para interpolar y editar imágenes desde w con cambios suaves y realistas.
- Entrenamiento más robusto y eficaz en datasets reducidos gracias a ADA.

Consistency GAN

Consistency GAN [54] introduce una integración entre el paradigma adversarial clásico y los principios de consistencia temporal desarrollados en modelos de difusión. El objetivo es aprovechar los beneficios de los modelos de consistencia —entrenados para mantener coherencia entre pasos de generación— dentro del marco GAN.

Concepto clave: El discriminador de Consistency GAN no sólo evalúa muestras únicas x , sino pares de muestras (x_t, x_{t+1}) generadas por un modelo consistente, penalizando incoherencias perceptuales entre pasos sucesivos. Esto fuerza al generador a mantener continuidad visual, ayudando a estabilizar el entrenamiento y mejorar la diversidad.

Ventajas:

- Mejora la coherencia entre muestras cercanas en el espacio latente.
- Reduce el colapso de modo al introducir supervisión adicional basada en transiciones.
- Compatible con arquitecturas GAN modernas como StyleGAN.

3.4.6. Valoración del enfoque GAN

Las variantes exploradas de las redes generativas adversariales han mostrado una evolución constante hacia mayor estabilidad, calidad visual y control semántico. Desde los primeros avances estructurales en DCGAN y la reformulación matemática de WGAN y WGAN-GP, hasta modelos expresivos como StyleGAN y extensiones modernas como Consistency GAN, cada arquitectura aporta beneficios específicos:

- **Estabilidad y convergencia:** WGAN-GP, LSGAN.
- **Transferencia de dominio:** CycleGAN.
- **Control sobre atributos:** StyleGAN, Consistency GAN.
- **Compatibilidad con pocos datos:** StyleGAN2-ADA.

En el contexto UAV, estas variantes permiten generar datos sintéticos más realistas, coherentes y controlables, lo cual resulta clave para tareas de entrenamiento supervisado, simulación visual y robustez frente a variaciones del entorno.

3.5. Modelos de Difusión (DDPM)

Los modelos de difusión, conocidos formalmente como Denoising Diffusion Probabilistic Models (DDPM), han emergido como una alternativa poderosa a GANs y VAEs, logrando resultados de vanguardia en generación de imágenes de alta calidad. Propuestos por Ho et al. [6], se basan en un proceso de destrucción progresiva de datos seguido por una reconstrucción aprendida paso a paso.

3.5.1. Fundamentos teóricos

El proceso consta de dos fases:

- **Difusión (forward process):** se añade ruido gaussiano al dato original x_0 durante T pasos para obtener $x_T \approx \mathcal{N}(0, I)$.
- **Denoising (reverse process):** se entrena una red neuronal para invertir el proceso paso a paso, es decir, para aproximar $q(x_{t-1}|x_t)$.

Cada paso de ruido es modelado como:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (3.18)$$

Y el objetivo de entrenamiento es minimizar:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \quad (3.19)$$

Donde ϵ_θ es la predicción del ruido por la red.

Comparativa con GANs: A diferencia de las GANs, los modelos de difusión:

- Tienen entrenamiento estable sin necesidad de discriminador.
- No sufren colapso de modo, cubriendo toda la distribución.
- Requieren muchos pasos de inferencia (coste computacional alto).

3.5.2. Arquitectura genérica de DDPM

La arquitectura base para los modelos de difusión utiliza una red neuronal *U-Net*, que incorpora información de tiempo mediante *embeddings* posicionales y emplea skip connections para facilitar el flujo de información desde baja hasta alta resolución.

La Figura 3.5.2 muestra un esquema simplificado de un paso de denoising en los modelos de difusión del tipo DDPM. En él, la entrada ruidosa x_t se procesa mediante una arquitectura *U-Net*, que recibe también como entrada la información temporal t codificada en forma de *embeddings* posicionales. La red predice el ruido $\epsilon_\theta(x_t, t)$ presente en la imagen, lo que permite invertir progresivamente el proceso de difusión y recuperar la muestra original.



Figura 3.5: Esquema simplificado de un paso de denoising en DDPM. Elaboración propia

3.5.3. Aplicaciones a UAVs

Los modelos de difusión han demostrado ser herramientas especialmente potentes para la generación de imágenes sintéticas de UAVs, debido a su capacidad para capturar distribuciones complejas y generar detalles visuales con alta fidelidad. Las principales aplicaciones en este ámbito incluyen:

- **Síntesis de UAVs con fondo complejo:** los DDPM son capaces de modelar tanto el dron como el fondo de manera coherente, incluso cuando existe variabilidad en textura o iluminación.
- **Interpolación temporal suave:** gracias a su estructura iterativa, permiten generar secuencias de UAVs con cambios progresivos de orientación o contexto.
- **Entrenamiento con pocos datos:** su estabilidad permite aprovechar pequeños conjuntos mediante reentrenamiento o transferencia.
- **Control detallado:** mediante técnicas como classifier-free guidance o conditioning, se pueden generar UAVs con atributos específicos (tipo de hélice, color, orientación).

Estas propiedades hacen de los modelos de difusión una solución ideal para generar datos de entrenamiento realistas, mejorar la robustez de sistemas de visión artificial embarcados, y explorar nuevos diseños morfológicos de UAVs en simulación.

3.5.4. Implementación básica en PyTorch

```

1 # Embedding de tiempo
2 class SinusoidalPosEmb(nn.Module):
3     def forward(self, t):
4         half_dim = self.dim // 2
5         emb = math.log(10000) / (half_dim - 1)
6         emb = torch.exp(torch.arange(half_dim) * -emb).to(t.device)
7         emb = t[:, None] * emb[None, :]
8         return torch.cat([emb.sin(), emb.cos()], dim=-1)
9
10 # Scheduler lineal beta
11 def make_beta_schedule(timesteps, beta_start=1e-4, beta_end=0.02):
12     return torch.linspace(beta_start, beta_end, timesteps)
13
14 # Bloque U-Net simplificado
15 class DenoiseBlock(nn.Module):
16     def __init__(self, channels, time_emb_dim):
17         self.time_mlp = nn.Sequential(
18             nn.Linear(time_emb_dim, channels),

```

```

19         nn.ReLU()
20     )
21     self.block = nn.Sequential(
22         nn.Conv2d(channels, channels, 3, padding=1),
23         nn.GroupNorm(8, channels),
24         nn.ReLU(),
25         nn.Conv2d(channels, channels, 3, padding=1)
26     )
27
28     def forward(self, x, t_emb):
29         time = self.time_mlp(t_emb).unsqueeze(-1).unsqueeze(-1)
30         return self.block(x + time)
31
32 class DDPM(nn.Module):
33     def __init__(self, channels=64, time_dim=128):
34         self.time_embed = SinusoidalPosEmb(time_dim)
35         self.net = nn.Sequential(
36             nn.Conv2d(1, channels, 3, padding=1),
37             DenoiseBlock(channels, time_dim),
38             DenoiseBlock(channels, time_dim),
39             nn.Conv2d(channels, 1, 3, padding=1)
40         )
41
42     def forward(self, x_t, t):
43         t_emb = self.time_embed(t)
44         return self.net[0](x_t), self.net[1](x_t, t_emb), self.net[2](x_t, t_emb), self.
net[3](x_t)

```

Listado 3.3: Implementación básica de un modelo DDPM

Este modelo representa una arquitectura DDPM simplificada, con embedding de tiempo, programación del ruido (scheduler) y bloques convolucionales que simulan U-Net. Puede extenderse fácilmente para incluir atención, capas condicionales o predicción de otros objetivos más allá del ruido.

Este modelo puede ampliarse a U-Nets con condicionamiento temporal y canales múltiples para tareas más complejas.

3.5.5. Variantes principales de modelos de difusión

En esta sección se describen las variantes más influyentes de los modelos DDPM, que han sido propuestas para mejorar aspectos clave como la velocidad de inferencia, la calidad visual, el control de atributos o la eficiencia de entrenamiento. Entre ellas se encuentran:

DDIM

Denoising Diffusion Implicit Models (DDIM), propuestos por Song et al. [55], ofrecen una alternativa al muestreo estocástico de DDPM mediante un enfoque determinista que permite acelerar la generación sin sacrificar la calidad de las muestras.

Motivación: DDPM requiere cientos o miles de pasos de muestreo, lo cual limita su aplicabilidad práctica. DDIM replantea el proceso de muestreo como una familia de trayectorias deterministas que preservan la calidad pero permiten usar menos pasos.

Proceso de muestreo: A partir de una muestra $x_T \sim \mathcal{N}(0, I)$, DDIM estima cada paso como:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1 - \alpha_t} \cdot \epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \eta^2} \cdot \epsilon_\theta(x_t, t) \quad (3.20)$$

Donde $\eta = 0$ implica muestreo puramente determinista, y valores mayores introducen estocasticidad.

Ventajas:

- Permite usar menos pasos de inferencia (25–50) manteniendo alta calidad.
- Conserva el marco de entrenamiento original (mismas redes y pérdidas).
- Apto para interpolación y edición latente gracias a su carácter determinista.

Improved DDPM

Improved DDPM, propuesto por Nichol y Dhariwal [40], extiende la formulación original de DDPM introduciendo varias mejoras clave que aumentan tanto la calidad como la eficiencia del modelo sin alterar la estructura general de entrenamiento.

Principales mejoras:

- **Predicción de parámetros múltiples:** el modelo aprende a predecir no solo el ruido ϵ , sino también la media y varianza del posterior, lo que permite mayor flexibilidad.
- **Scheduler ajustado:** uso de calendarios de β_t no lineales (coseno, cuadrático) para una mejor distribución de ruido.

- **Objetivo alternativo:** introduce una combinación ponderada de pérdidas sobre x_0 y ϵ , mejorando la fidelidad.

Ventajas:

- Mejora sustancial de FID frente al DDPM original.
- Mayor flexibilidad para condicionar la generación.
- Compatible con arquitecturas y flujos de entrenamiento existentes.

FDM

El modelo Fast Diffusion Model (FDM), propuesto por Salimans y Ho [56], busca reducir drásticamente los pasos de inferencia necesarios en los modelos de difusión, manteniendo al mismo tiempo una alta calidad visual. Se basa en reentrenar un modelo DDPM estándar con una secuencia de pasos de muestreo más corta y adaptada.

Motivación: La principal limitación práctica de los DDPM es la lentitud de muestreo. FDM emplea un enfoque basado en schedule distillation: el modelo es progresivamente adaptado a funcionar con menos pasos, aprendiendo una nueva secuencia de ruido efectiva.

Funcionamiento: Se parte de un modelo entrenado con muchos pasos y se reentrena utilizando trayectorias más cortas. La red se ajusta para aprender directamente el mapeo entre x_T y x_0 en un número reducido de iteraciones (8–16 pasos).

Ventajas:

- Reducción significativa del tiempo de muestreo (hasta 20×).
- Preserva alta calidad visual y diversidad sin recurrir a nuevas arquitecturas.
- Compatible con métodos de guidance y condicionamiento.

Latent Diffusion Models

Latent Diffusion Models (LDM), propuestos por Rombach et al. [7], combinan la potencia de los modelos de difusión con la eficiencia de operar en espacios latentes

comprimidos. En lugar de aplicar el proceso de ruido directamente sobre las imágenes, lo hacen sobre una representación latente aprendida mediante un autoencoder convolucional.

Motivación: Generar imágenes directamente en alta resolución (por ejemplo, 512x512) con DDPM es costoso en términos de memoria y cómputo. LDM reduce este coste al aplicar el modelo sobre una versión comprimida y semánticamente rica de las imágenes.

Funcionamiento:

- Se entrena un autoencoder E para mapear imágenes $x \rightarrow z$, y un decodificador D para reconstruir $x \leftarrow z$.
- El modelo de difusión opera sobre z , generando z_0 a partir de ruido, y luego $x_0 = D(z_0)$.

Ventajas:

- Permite generación en resoluciones altas con menos recursos.
- Mejora la velocidad de entrenamiento y muestreo.
- Compatible con condicionamiento textual o visual mediante técnicas como cross-attention.

Stable Diffusion

Stable Diffusion [7] es una implementación popular y eficiente del paradigma Latent Diffusion, que ha sido ampliamente adoptada por su capacidad para generar imágenes de alta resolución condicionadas por texto. Se trata de una arquitectura entrenada sobre grandes corpus de texto e imagen, y optimizada para ejecutarse en GPUs de consumo mediante el uso de espacios latentes comprimidos.

Motivación: Stable Diffusion busca combinar la calidad de modelos como Imagen o DALL·E 2 con una eficiencia accesible para usuarios con hardware limitado. Se fundamenta en los Latent Diffusion Models (LDM), pero introduce mejoras en el condicionamiento textual, el entrenamiento y la arquitectura del decodificador.

Funcionamiento:

- Utiliza un autoencoder entrenado para mapear imágenes al espacio latente $z \in \mathbb{R}^{4 \times 64 \times 64}$.
- Aplica un modelo de difusión sobre z , condicionado por embeddings de texto obtenidos con CLIP o BERT.
- Reconstruye la imagen final mediante un decodificador convolucional $D(z)$.

Ventajas:

- Alta calidad visual con uso eficiente de memoria (VRAM).
- Control mediante prompts textuales precisos.
- Arquitectura abierta y fácilmente personalizable.

EDM (Elucidated Diffusion Models)

Elucidated Diffusion Models (EDM), propuestos por Karras et al. [57], representan un marco teórico y práctico refinado para entrenar modelos de difusión, mejorando la eficiencia, estabilidad y calidad de generación sin alterar radicalmente la estructura básica de los DDPM.

Motivación: Los autores buscan una formulación unificada que permita entender y optimizar diferentes estrategias de entrenamiento en difusión, explicando cuándo y por qué ciertas configuraciones producen mejores resultados. EDM también introduce mejoras técnicas específicas que conducen a rendimientos superiores en benchmarks visuales.

Aportaciones principales:

- Reformulación del proceso de muestreo como *variance preserving SDEs* con sampling por second-order solvers.
- Uso de un rango óptimo de escalado de ruido $\sigma \in [\sigma_{min}, \sigma_{max}]$ adaptado a los datos.
- Introducción del loss *v-prediction* que mejora la estabilidad y generalización.

Ventajas:

- Mejora la calidad de generación (FID) frente a DDPM y DDIM.

- Reduce la sensibilidad a la configuración del scheduler.
- Generaliza múltiples variantes de entrenamiento bajo un marco común.

Imagen

Imagen [58], desarrollado por Google Research, es un modelo de difusión de última generación que ha establecido nuevos estándares en calidad fotorrealista para la generación de imágenes condicionadas por texto. Se basa en un esquema de generación jerárquica y escalonada, empleando redes de difusión en resolución creciente.

Motivación: El objetivo de Imagen es maximizar la fidelidad perceptual y semántica de las imágenes generadas, combinando embeddings textuales avanzados con una arquitectura de difusión progresiva que supera las limitaciones de resolución de los modelos convencionales.

Funcionamiento:

- Utiliza embeddings del modelo de lenguaje T5-XXL para codificar el texto.
- Aplica un modelo de difusión sobre imágenes de baja resolución (64x64).
- Realiza superresolución escalonada hasta alcanzar 1024x1024 usando modelos de difusión adicionales.

Ventajas:

- Estado del arte en calidad visual y coherencia semántica.
- Manejo robusto de relaciones espaciales complejas y estructuras detalladas.
- Arquitectura modular que facilita adaptación a distintas resoluciones.

Consistency Models

Los Consistency Models (CM), introducidos por Song et al. [59], reformulan el paradigma de los modelos de difusión para permitir una generación de imágenes en una sola etapa de inferencia, manteniendo una alta calidad visual sin necesidad de recorrer muchos pasos de muestreo. Esto representa un cambio radical respecto a los DDPM tradicionales.

Motivación: Reducir el tiempo de generación al mínimo posible, eliminando la necesidad de múltiples pasos iterativos, sin perder las ventajas de estabilidad y diversidad de los modelos de difusión.

Funcionamiento:

- Se entrena una red de consistencia $f(x_t, t)$ para garantizar que $f(f(x_t, t + s), t) \approx f(x_t, t)$, es decir, que la red mantenga consistencia temporal entre pasos.
- El modelo aprende un operador consistente que aproxima directamente el resultado final desde una entrada ruidosa inicial.

Ventajas:

- Generación en un único paso (one-step sampling) o muy pocos pasos (2–4).
- Reducción masiva del tiempo de inferencia sin pérdida sustancial de calidad.
- Compatible con guidance textual y condicional.

3.5.6. Valoración del enfoque modelos de difusión

Los modelos de difusión han supuesto un punto de inflexión en generación sintética, al combinar una elevada fidelidad visual con un entrenamiento estable y predecible. Frente a las GANs, que suelen requerir un balance delicado entre generador y discriminador, los DDPM permiten aproximar de forma robusta distribuciones complejas sin colapso de modo ni oscilaciones inestables.

En esta sección se han abordado tanto los fundamentos del esquema DDPM clásico como las principales variantes propuestas en los últimos años. Las mejoras han seguido líneas claras: acelerar la inferencia (DDIM, FDM, Consistency Models), reducir el coste computacional (Latent Diffusion, Stable Diffusion) o incrementar la capacidad de control (Imagen, EDM).

- **Aceleración:** DDIM permite muestreo determinista en pocos pasos; FDM y Consistency recortan aún más la latencia, con trayectorias optimizadas o un único paso.
- **Eficiencia:** modelos como LDM y Stable Diffusion trasladan la generación al espacio latente, reduciendo el uso de memoria y acelerando el entrenamiento.

- **Calidad:** EDM e Imagen empujan el FID al mínimo conocido, siendo referencia actual en tareas text-to-image de alta resolución.

En el contexto específico de UAVs, las ventajas son evidentes: posibilidad de generar escenas completas desde prompts (Stable Diffusion), control sobre condiciones (ángulo, entorno, hora del día), y síntesis rápida de ejemplos diversos para entrenar detectores robustos. Algunas variantes permiten además despliegue en entornos embarcados, abriendo la puerta a generación en línea durante misiones reales.

3.6. Comparativa de las arquitecturas generativas

Con el análisis de VAE, GAN y DDPM ya completado, es posible establecer una comparativa entre estas tres familias, considerando aspectos clave como calidad visual, diversidad de resultados, estabilidad en el entrenamiento y capacidad de control. La Tabla 3.2 sintetiza estos aspectos de forma cualitativa.

Modelo	Calidad visual	Diversidad	Estabilidad	Control condicional
VAE	Media	Alta	Muy alta	Medio
GAN	Alta	Media	Baja	Medio
DDPM	Muy alta	Muy alta	Alta	Alta

Tabla. 3.2: Comparativa general entre VAE, GAN y DDPM.

Aplicación en UAVs

- **VAE** permite codificar UAVs en espacios latentes compactos, útiles para tareas de compresión, reconstrucción estructurada y detección de anomalías.
- **GAN** ofrece inferencia prácticamente instantánea, siendo útil en pipelines donde se requiera gran volumen de datos con apariencia realista y atributos específicos.
- **DDPM** destaca en fidelidad visual y diversidad, siendo la opción preferente cuando se requiere alta calidad o condicionamiento fino (por ejemplo, generación desde texto o mapas semánticos).

Cada arquitectura cubre un nicho distinto dentro del pipeline de generación UAV, y en muchos casos pueden integrarse de forma complementaria (por ejemplo, usando

VAE como codificador, GAN para preentrenamiento rápido, y DDPM para refinamiento). La elección final depende de los recursos disponibles, el volumen de datos y el tipo de aplicación objetivo (entrenamiento, simulación, validación, etc.).

Capítulo 4

Metodología y Experimentación

4.1. Introducción

Este capítulo detalla el enfoque metodológico y el diseño experimental empleados para evaluar los modelos generativos seleccionados en el contexto de la generación de imágenes de objetos en vuelo. Tras definir los objetivos y los recursos disponibles, se expone el flujo de trabajo seguido: desde la selección y preparación de los datos, pasando por la implementación y ajuste de cada arquitectura, hasta la obtención de métricas cuantitativas y cualitativas. El propósito es garantizar la reproducibilidad de los experimentos y fundamentar las conclusiones en resultados contrastados.

En primer lugar, se presenta una visión global del proceso (Figura 4.1), organizada en cinco fases principales: revisión bibliográfica, diseño experimental, implementación de modelos, experimentación y análisis de resultados. A continuación, se describen los recursos utilizados (hardware y software), los criterios de búsqueda y los procedimientos de generación y partición de los conjuntos de datos. Seguidamente, se especifican los protocolos de entrenamiento para cada familia de modelos (VAE, GAN y DDPM), incluyendo hiperparámetros, estrategias de optimización y mecanismos de control de calidad. Por último, se definen las métricas objetivas y subjetivas empleadas para evaluar la fidelidad, diversidad y eficiencia de las imágenes sintéticas, así como las herramientas de registro y trazabilidad que garantizan la transparencia del proceso experimental.

4.2. Visión general del flujo de trabajo

El desarrollo metodológico y experimental de este TFG se estructura en cinco fases interrelacionadas, que abarcan desde la revisión bibliográfica inicial hasta el análisis final de resultados. Cada etapa cuenta con entregables definidos y criterios de éxito específicos, lo que garantiza la trazabilidad y la reproducibilidad a lo largo de todo el proceso. Este flujo de trabajo se resume visualmente en la Figura 4.1, donde se representa la secuencia lógica que guía la progresión del proyecto.

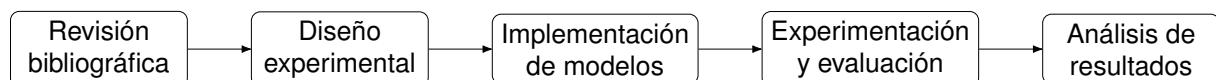


Figura 4.1: Flujo de trabajo general: de la revisión bibliográfica al análisis de resultados.

Durante la fase de *revisión bibliográfica* se identificaron los enfoques, métricas y herramientas más relevantes (GAN, VAE, DDPM, FDM, StyleGAN-ADA, entre otros). A partir de esta base, el *diseño experimental* definió los datasets utilizados, los esquemas de partición y los protocolos de evaluación. La fase de *implementación de modelos* incluyó la adaptación de repositorios oficiales y el desarrollo de scripts específicos para el entrenamiento. Posteriormente, en la etapa de *experimentación y evaluación*, se ejecutaron los entrenamientos, se recogieron métricas objetivas (FID, IS, SSIM, PSNR) y se realizaron pruebas subjetivas. Finalmente, en el *análisis de resultados* se interpretaron los datos obtenidos y se contrastaron las hipótesis planteadas, estableciendo así las bases para la discusión crítica y las conclusiones del trabajo.

4.3. Diseño experimental

La fase de diseño experimental constituye el núcleo organizativo del trabajo, al estructurar cómo se compararán los distintos modelos generativos bajo condiciones controladas y homogéneas. Este diseño se ha elaborado considerando las limitaciones computacionales, la diversidad arquitectónica de los modelos revisados y los objetivos específicos del estudio: generar imágenes sintéticas de objetos en vuelo con alta fidelidad visual, diversidad estructural y controlabilidad semántica.

4.3.1. Criterios de selección de modelos

El conjunto de modelos seleccionados representa una muestra equilibrada y representativa de las tres grandes familias de generadores: **Autoencoders Variacionales (VAE)**, **Redes Generativas Adversariales (GAN)** y **Modelos de Difusión (DDPM)**. La elección se fundamentó en los siguientes criterios:

- Relevancia científica y número de citas en la literatura especializada.
- Disponibilidad de implementaciones abiertas, mantenidas y documentadas.
- Viabilidad de entrenamiento en entornos de cómputo locales.
- Cobertura de variantes relevantes dentro de cada familia.

Los modelos finalmente implementados fueron:

- **VAE**: β -VAE, InfoVAE, WAE-MMD.
- **GAN**: DCGAN, WGAN, WGAN-GP, StyleGAN2-ADA.
- **DDPM**: FDM (*Fast Diffusion Models*) sobre EDM VP y VE.

4.3.2. Hipótesis de trabajo

Con el objetivo de evaluar el rendimiento y la utilidad práctica de cada arquitectura, se formularon las siguientes hipótesis:

1. Los modelos de difusión (FDM) alcanzarán la mayor calidad visual según FID y pruebas subjetivas, a cambio de mayores tiempos de generación.
2. Las GAN ofrecerán un equilibrio favorable entre realismo visual y velocidad de inferencia, especialmente en arquitecturas modernas como StyleGAN2-ADA.
3. Los VAE, aunque teóricamente robustos, generarán imágenes menos detalladas pero con una latencia más interpretable y aprovechable para edición.

4.3.3. Variables y configuración experimental

El diseño factorial se estructura en torno a tres dimensiones principales:

- **Modelo generativo:** ocho arquitecturas distintas.
- **Tamaño de entrada:** imágenes redimensionadas a 64×64 píxeles.
- **Evaluación cruzada:** cada experimento se repitió al menos tres veces con semillas distintas para comprobar su robustez.

4.3.4. Esquema del flujo de trabajo

La Figura 4.2 resume gráficamente el flujo metodológico implementado para la comparación entre modelos. Se parte de un conjunto base de datos reales y se lleva a cabo el entrenamiento individualizado de cada arquitectura. A continuación, se generan imágenes sintéticas que se someten a evaluación objetiva (FID, IS, SSIM, PSNR). Los resultados se consolidan posteriormente para su análisis estadístico y visual.

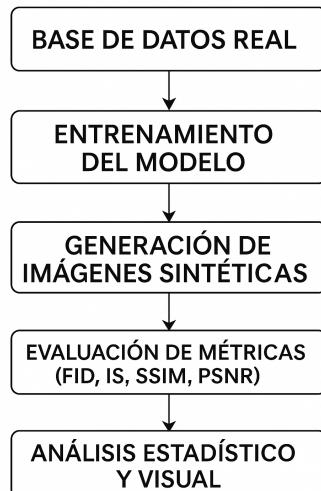


Figura 4.2: Esquema general del flujo de trabajo experimental. Elaboración propia

4.4. Recursos computacionales y herramientas

El desarrollo de este trabajo ha requerido una combinación equilibrada entre bibliotecas especializadas de aprendizaje profundo, entornos de desarrollo eficientes

y recursos computacionales diversos, adecuados para el entrenamiento, evaluación y comparación de modelos generativos de última generación. A continuación se detallan los sistemas utilizados a lo largo de las distintas fases del proyecto.

4.4.1. Hardware utilizado

Dada la elevada carga computacional asociada al entrenamiento de modelos como StyleGAN2-ADA o FDM-EDM, se ha trabajado con distintos entornos físicos, optimizados para diferentes momentos del flujo experimental:

- **Sistema DGX-1 (entorno cedido temporalmente):**
 - **GPU:** 8 × NVIDIA Tesla V100 (32 GB cada una)
 - **RAM:** 512 GB
 - **Uso:** Entrenamiento distribuido y pruebas finales de StyleGAN2-ADA, FDM y VAE sobre los conjuntos CelebA y UAV.

Toda la experimentación cuyos resultados se presentan en el Capítulo 5 fue ejecutada exclusivamente en este entorno, lo que permitió garantizar consistencia en las métricas, reproducibilidad en las ejecuciones y tiempos de convergencia adecuados para modelos de gran escala.

- **Estación local A:**
 - **CPU:** Intel Core i9-11900k
 - **GPU:** NVIDIA RTX 3070 Ti (8 GB VRAM)
 - **RAM:** 32 GB DDR4
 - **Almacenamiento:** SSD NVMe de 1 TB
 - **Sistema operativo:** Linux Mint Cinnamon 22.1

Utilizada para pruebas preliminares, validación de scripts y análisis exploratorio.

- **Estación local B:**
 - **CPU:** AMD Ryzen 7 9800X3D
 - **GPU:** NVIDIA RTX 4060 Ti (16 GB VRAM)
 - **RAM:** 32 GB DDR4
 - **Sistema operativo:** Linux Mint Cinnamon 22.1

Plataforma empleada para tareas de depuración, pruebas ligeras y gestión de visualizaciones.

4.4.2. Entornos de desarrollo

Todos los modelos han sido implementados en **Python 3.12** empleando la biblioteca **PyTorch 2.7.1**. El entorno de trabajo principal ha sido VSCode, junto con gestores de entorno (conda, pip) y control de versiones mediante Git. La documentación, gráficos y escritura del TFG se han realizado en **LaTeX**, incluyendo diagramas con TikZ y PGFPlots.

4.4.3. Gestión de experimentos

Para facilitar la reproducibilidad, cada experimento se ha registrado con parámetros de configuración, versiones de modelo y métricas obtenidas. Se ha utilizado TensorBoard para visualizar métricas de entrenamiento, y scripts propios para automatizar la evaluación cuantitativa (FID, IS, PSNR, SSIM) y la generación de comparativas visuales.

Parámetros de entrenamiento. Los valores seleccionados para los principales hiperparámetros se han mantenido constantes dentro de cada combinación arquitectura/dominio, de modo que los resultados fueran comparables y replicables. A continuación se resumen los ajustes utilizados:

- **GANs (DCGAN, WGAN, StyleGAN2-ADA, etc.):**

- **CelebA:** 20 epochs
- **UAV:** 100 epochs
- **Learning rate:** entre $1 \cdot 10^{-4}$ y $2 \cdot 10^{-5}$

- **VAEs (Beta-VAE, WAE-MMD, InfoVAE):**

- **CelebA:** 20 epochs
- **UAV:** 100 epochs
- **Learning rate:** entre $1 \cdot 10^{-4}$ y $2 \cdot 10^{-5}$

- **Modelos de difusión (FDM-VP, FDM-EDM):**

- **CelebA:** 1.000 kimg
- **UAV:** 10.000 kimg
- **Learning rate:** $2 \cdot 10^{-5}$

Espacio latente. Todos los modelos comparten una dimensión latente de \mathbb{R}^{128} , seleccionada como punto intermedio entre capacidad representacional y estabilidad durante el entrenamiento.

Tamaño de lote (batch size). Se ha fijado un tamaño de lote de 64 para GANs y VAEs. En el caso de los modelos de difusión, se ha ajustado dinámicamente en función de la VRAM disponible, con valores típicos entre 8 y 32.

Hiperparámetros por defecto. Salvo los valores indicados anteriormente, se han mantenido los parámetros propuestos por los artículos originales de cada arquitectura. Esto incluye funciones de activación, inicializaciones, regularización, coeficientes de pérdida o configuraciones del optimizador (Adam, $\beta_1 = 0,5$, $\beta_2 = 0,999$ en GANs, etc.).

Control de reproducibilidad. Cada experimento ha sido inicializado con una semilla fija mediante `torch.manual_seed()`, asegurando la reproducibilidad.

4.4.4. Datos

En este trabajo se han utilizado dos conjuntos de datos principales: uno de carácter público y estandarizado (CelebA), y otro de elaboración propia construido a partir de recursos obtenidos en la plataforma Kaggle. Ambos han sido empleados tanto para entrenar modelos generativos como para evaluar el impacto de los datos sintéticos generados.

Conjunto CelebA

El conjunto CelebFaces Attributes (CelebA) es un dataset ampliamente utilizado en el campo de la generación de imágenes, particularmente en tareas de modelado facial. Contiene más de 200.000 imágenes de rostros humanos con anotaciones de atributos faciales, cubriendo variabilidad en género, expresión, orientación y condiciones de iluminación.

Para este trabajo se ha empleado una versión recortada y centrada del conjunto, redimensionando todas las imágenes a una resolución fija de 64×64 píxeles y eliminando etiquetas adicionales no relevantes para la tarea. El conjunto ha sido dividido

en subconjuntos de entrenamiento (80 %), validación (10 %) y prueba (10 %) de forma aleatorizada solo para los modelos que realizan reconstrucción.

Conjunto propio basado en Kaggle

El segundo conjunto ha sido construido manualmente a partir de múltiples datasets temáticamente relacionados con objetos voladores, recopilados desde la plataforma Kaggle (véase Sección 4.4). El proceso de integración y limpieza ha incluido la eliminación de imágenes duplicadas, el filtrado por calidad mínima (resoluciones inferiores a 96×96 fueron descartadas) y la homogeneización del formato de color.

El conjunto resultante consta de un total de 16.000 imágenes y presenta una gran diversidad visual en cuanto a fondo, ángulo, distancia y contexto. Para aumentar su representatividad y robustez frente a sobreajuste, se aplicaron técnicas de aumento de datos tales como rotación aleatoria, traslación, inversión horizontal y cambios de brillo y saturación. Estas transformaciones se aplicaron exclusivamente durante el entrenamiento para preservar la validez de los conjuntos de validación y test.

Preprocesamiento y estructura común

Ambos conjuntos han sido sometidos a un proceso homogéneo de preprocesamiento:

- Redimensionamiento a 64×64 píxeles con preservación de aspecto mediante recorte central.
- Normalización de valores de píxel al rango $[-1, 1]$ para facilitar el entrenamiento de redes neuronales.
- Conversión a formato RGB unificado para garantizar la compatibilidad en las fases de generación.

Muestra visual y análisis comparativo

En la Figura 4.3 se presenta una muestra visual representativa de ambos conjuntos de datos utilizados. La parte izquierda del collage muestra imágenes extraídas del



Figura 4.3: Collage comparativo. Izquierda: subconjunto de CelebA. Derecha: subconjunto del dataset propio de drones.

conjunto CelebA, mientras que la parte derecha contiene ejemplos provenientes del conjunto propio construido a partir de datasets de Kaggle.

Visualmente, se aprecian diferencias estructurales notables entre ambos conjuntos. CelebA se caracteriza por contener retratos frontales y centrados, con condiciones lumínicas y composiciones relativamente homogéneas. Las caras suelen estar bien delimitadas, con fondos desenfocados y una estética coherente.

Por el contrario, el conjunto propio presenta una mayor heterogeneidad visual. Las imágenes muestran objetos voladores en distintas orientaciones, escalas y contextos. Algunos están parcialmente ocluidos, en escenarios urbanos o naturales, con variaciones importantes en brillo, contraste y complejidad de fondo. Esta diversidad impone retos adicionales a los modelos generativos, ya que deben aprender a sintetizar estructuras más variadas y resolver ambigüedades espaciales o texturales más frecuentes.

Esta comparación visual justifica tanto la necesidad de modelos robustos como la conveniencia de técnicas de aumentación y normalización para homogeneizar parcialmente la distribución de entrada.

4.4.5. Entorno de desarrollo y bibliotecas

La implementación de los modelos y la ejecución de los experimentos se han llevado a cabo principalmente en Python 3.12. A nivel de frameworks y librerías, se emplearon las siguientes herramientas:

- **PyTorch 2.7.1 [60]**: framework principal para definición de modelos y entrenamiento.
- **Torchvision**: soporte para transformaciones, cargas y visualización de datos.
- **scikit-learn & NumPy**: operaciones estadísticas y cálculo de métricas auxiliares.
- **Matplotlib & Seaborn**: representación gráfica de resultados y comparativas.

Los entornos de ejecución fueron gestionados mediante entornos virtuales (`conda`), garantizando la reproducibilidad y el aislamiento de dependencias entre proyectos.

4.4.6. Control de versiones y documentación

El código fuente del proyecto fue gestionado con **Git**, utilizando un repositorio privado en GitHub. Cada experimento se documentó en scripts parametrizados y comentados, lo que permitió reproducir configuraciones previas, comparar resultados y automatizar tareas rutinarias.

En síntesis, la combinación de una infraestructura local robusta con herramientas modernas de desarrollo ha permitido realizar el entrenamiento, evaluación y análisis de los modelos generativos propuestos de forma eficiente, trazable y controlada.

4.5. Modelos seleccionados

Para llevar a cabo la experimentación, se ha seleccionado un conjunto representativo de modelos pertenecientes a cada una de las familias generativas abordadas en este trabajo: *Autoencoders Variacionales* (VAE), *Redes Generativas Adversariales* (GAN) y *Modelos de Difusión* (DDPM). La elección se ha fundamentado tanto en la relevancia de cada arquitectura en la literatura como en su disponibilidad y viabilidad de implementación.

4.5.1. Modelos GAN

Dentro del grupo de las GAN, se ha incluido un abanico de variantes que cubren desde aproximaciones clásicas hasta propuestas recientes orientadas a mejorar la estabilidad del entrenamiento y la calidad visual:

- **DCGAN [46]**: modelo pionero que estabilizó el entrenamiento de GANs mediante arquitecturas convolucionales profundas.
- **WGAN [47]**: reformula la función de pérdida mediante la distancia de Wasserstein para mejorar la convergencia.
- **WGAN-GP [48]**: introduce una penalización en el gradiente para evitar el recorte de pesos y mejorar la estabilidad.
- **StyleGAN2-ADA [52]**: arquitectura de última generación que aplica regularización adaptativa para permitir el entrenamiento con datasets reducidos.

Estas variantes permiten comparar el impacto de distintas estrategias de entrenamiento, regularización y control estilístico.

4.5.2. Modelos VAE

En cuanto a autoencoders variacionales, se han elegido variantes que introducen mejoras sobre el modelo original en cuanto a regularización y estructura latente:

- **Beta-VAE [41]**: promueve representaciones latentes disentangled mediante una penalización aumentada sobre la divergencia KL.
- **InfoVAE [42]**: optimiza una variante del ELBO que maximiza la información mutua entre la variable latente y los datos.
- **WAE-MMD [43]**: introduce una regularización mediante *Maximum Mean Discrepancy* para mejorar la alineación del espacio latente.

Estas variantes permiten evaluar cómo diferentes formulaciones influyen en la calidad y control de la generación.

4.5.3. Modelos de Difusión

Para los modelos de difusión, se ha optado por una implementación moderna optimizada para eficiencia computacional:

- **FDM (Fast Diffusion Models)**: arquitectura que emplea trayectorias deterministas tipo DDIM [55], lo que permite reducir significativamente el número de pasos de inferencia sin comprometer la calidad visual. En este trabajo se ha implementado FDM sobre tres variantes del esquema de ruido: **EDM (Elucidated Diffusion Models)**, **VP (Variance Preserving)** y **VE (Variance Exploding)**, evaluando su rendimiento comparativo en los dominios CelebA y UAV.

Este modelo permite aprovechar los beneficios de la difusión manteniendo una viabilidad práctica en entornos con recursos limitados.

4.5.4. Resumen de modelos

La Tabla 4.1 resume las arquitecturas evaluadas y su clasificación por familia:

Modelo	Familia
DCGAN	GAN
WGAN	GAN
WGAN-GP	GAN
StyleGAN2-ADA	GAN
Beta-VAE	VAE
InfoVAE	VAE
WAE-MMD	VAE
FDM	Difusión (DDPM)

Tabla. 4.1: Modelos generativos seleccionados para la experimentación.

4.5.5. Procedimiento experimental

El procedimiento experimental adoptado se ha estructurado en las siguientes fases secuenciales:

1. **Preparación de los conjuntos de datos:** los datos fueron descargados, organizados y redimensionados conforme a los requerimientos de entrada de cada modelo. Se estandarizó la resolución a 64×64 píxeles. En el conjunto UAV se aplicaron técnicas de limpieza y filtrado; en CelebA se aplicó únicamente redimensionado y normalización.
2. **Entrenamiento de los modelos:** cada arquitectura se entrenó de forma independiente sobre ambos conjuntos de datos (UAV y CelebA). Los hiperparámetros (épocas, tamaño de lote, tasa de aprendizaje) se ajustaron mediante validación cruzada simple sobre un subconjunto del conjunto de entrenamiento. Se monitorizó la convergencia mediante las pérdidas específicas de cada modelo y visualizaciones periódicas.
3. **Evaluación cuantitativa:** tras el entrenamiento, se evaluó cada modelo en términos de calidad visual y fidelidad estadística usando las métricas FID, IS, SSIM y PSNR. Para FID e IS se generaron 5.000 imágenes por modelo, comparadas con subconjuntos de imágenes reales no vistas. SSIM y PSNR se calcularon únicamente en arquitecturas reconstructivas.
4. **Análisis cualitativo:** además de las métricas numéricas, se realizó una inspección visual sobre muestras seleccionadas, evaluando coherencia semántica, diversidad y presencia de artefactos. Los resultados se discuten en el Capítulo 5.

Este procedimiento garantiza una comparación equitativa y sistemática entre los modelos seleccionados, minimizando la influencia de factores externos y centrándose en la capacidad generativa de cada arquitectura.

Capítulo 5

Resultados

Este capítulo presenta un análisis exhaustivo de los resultados obtenidos tras aplicar diversos modelos generativos sobre dos conjuntos de datos con características complementarias: **CelebA**, formado por rostros humanos centrados, y **UAV Drones**, que contiene imágenes de vehículos aéreos no tripulados desde múltiples perspectivas y fondos.

El objetivo de este apartado es doble: por un lado, evaluar de forma objetiva la calidad de las imágenes generadas mediante distintas métricas estándar en el campo de la síntesis de imágenes; por otro, establecer una correlación clara entre el diseño del modelo, su comportamiento durante el entrenamiento y las cualidades visuales de las imágenes resultantes.

El capítulo se divide en dos bloques principales:

1. La **reconstrucción** de imágenes, en la que se parte de una imagen real que se codifica y decodifica mediante un modelo VAE. Aquí se evalúa la capacidad del modelo para comprimir la información sin pérdida sustancial de fidelidad visual ni estructural.
2. La **generación pura** de imágenes desde ruido aleatorio, utilizando tanto arquitecturas GAN tradicionales como modelos de difusión (DDPM). Esta tarea refleja la habilidad del modelo para sintetizar muestras convincentes sin referencia directa.

5.1. Metodología de evaluación

Para evaluar la calidad y utilidad de las imágenes generadas por las arquitecturas estudiadas, se ha seguido una estrategia mixta basada en métricas objetivas ampliamente aceptadas en la literatura, complementadas con análisis visuales cualitativos. Esta combinación permite comparar modelos de forma reproducible y cuantificable, sin renunciar a la interpretación empírica de los resultados.

5.1.1. Métricas utilizadas

Las siguientes métricas han sido calculadas para cada experimento, cubriendo aspectos clave como fidelidad visual, diversidad estadística, similitud perceptual y eficiencia computacional:

- **Frechet Inception Distance (FID)** [61]: mide la distancia entre las distribuciones estadísticas (media y covarianza) de activaciones extraídas por una red Inception v3, calculadas sobre imágenes reales y generadas. Valores más bajos indican mayor similitud.
- **Inception Score (IS)** [62]: evalúa simultáneamente calidad y diversidad a partir de la entropía de las predicciones de la red Inception. Valores altos reflejan imágenes nítidas y variadas.
- **Peak Signal-to-Noise Ratio (PSNR)** [63]: mide la relación entre la señal máxima y el error cuadrático medio respecto a una imagen de referencia. Se emplea en tareas de reconstrucción. Cuanto mayor, mejor.
- **Structural Similarity Index (SSIM)** [64]: estima la similitud estructural entre dos imágenes teniendo en cuenta luminancia, contraste y textura. Presenta mayor correlación con la percepción humana que PSNR.
- **Tiempo de entrenamiento:** incluido como métrica de eficiencia práctica. Permite valorar el coste computacional relativo entre arquitecturas bajo condiciones experimentales similares.

5.1.2. Procedimiento experimental

Todas las métricas se calcularon sobre un conjunto común de vectores latentes z , generados con la misma semilla para asegurar condiciones comparables entre modelos. En el caso de PSNR y SSIM, se emplearon imágenes reales como referencia directa en modelos reconstructivos (como VAE o FDM). Para FID e IS, las imágenes generadas se compararon con un subconjunto del dataset real de evaluación, preprocesado a la misma resolución (64×64) y normalización.

Las implementaciones de FID e IS utilizadas están validadas por la comunidad y basadas en PyTorch y SciPy; PSNR y SSIM se calcularon mediante funciones estándar del paquete scikit-image. El tiempo de entrenamiento se registró mediante temporización directa por arquitectura.

5.1.3. Evaluación cualitativa

Además del análisis numérico, se ha realizado una valoración cualitativa detallada a partir de figuras comparativas. Se han seleccionado ejemplos representativos que muestran imágenes reales junto con sus respectivas reconstrucciones (en modelos como VAE y FDM) o generaciones libres (en GAN y DDPM). Esta comparación visual se ha guiado por criterios de:

- Fidelidad perceptual (nitidez, textura, color)
- Coherencia estructural (contornos, proporciones)
- Consistencia del fondo y elementos semánticos
- Diversidad morfológica entre muestras generadas

Este enfoque permite interpretar los resultados cuantitativos desde una perspectiva empírica, revelando matices que las métricas objetivas pueden pasar por alto.

5.1.4. Consideraciones y limitaciones

A pesar de su utilidad, las métricas empleadas presentan limitaciones importantes. FID e IS están basadas en una red entrenada en ImageNet, lo que introduce sesgos cuando se evalúan dominios distintos, como imágenes técnicas o aéreas. PSNR y

SSIM requieren correspondencia exacta entre imagen generada y referencia, lo que restringe su aplicabilidad a arquitecturas reconstructivas.

Por otro lado, el tiempo de entrenamiento depende de múltiples factores (batch size, GPU, implementación), por lo que solo se ha interpretado como una estimación relativa entre modelos bajo un entorno controlado.

5.1.5. Síntesis comparativa

A lo largo del capítulo se analizará cómo las características propias de cada arquitectura —como la regularización en VAE, el tipo de convoluciones y normalización en GAN, o el número de pasos de muestreo en DDPM— influyen en la calidad de las imágenes generadas. Se pondrá especial atención a los artefactos frecuentes, deformaciones estructurales, homogeneidad de fondo y nitidez perceptual.

El capítulo concluye con una síntesis entre los resultados obtenidos en distintos dominios (rostros y UAVs), y una reflexión crítica sobre las limitaciones de generalización y los retos futuros en generación sintética eficiente y controlable.

5.2. Reconstrucción en CelebA con modelos VAE

La primera parte del estudio se centra en la evaluación de la capacidad reconstructiva de distintos modelos de codificación variacional aplicados al conjunto de datos CelebA. Este conjunto, compuesto por imágenes de rostros humanos en diversas expresiones, poses y condiciones de iluminación, proporciona un entorno ideal para comprobar el equilibrio entre fidelidad de reconstrucción, compresión latente y robustez de generalización en entornos de alta estructura semántica.

5.2.1. Evaluación cuantitativa

La [Tabla 5.1](#) recoge los resultados cuantitativos obtenidos para los tres modelos de tipo VAE considerados: WAE-MMD, InfoVAE y Beta-VAE. Las métricas empleadas incluyen:

- **FID** (Fréchet Inception Distance): cuanto menor, mejor similitud estadística con el conjunto real.
- **IS** (Inception Score): valora diversidad y confianza de las clases predichas.
- **PSNR** (Peak Signal-to-Noise Ratio): sensibilidad a diferencias de pixel; mide fielidad directa.
- **SSIM** (Structural Similarity Index): valora coherencia estructural.
- **Tiempo**: duración del entrenamiento del modelo (H:M:S).

Modelo	FID ↓	IS ↑	PSNR ↑	SSIM ↑	Tiempo
WAE-MMD	122.90	2.65	21.79	0.621	01:48:37
InfoVAE	139.07	2.51	20.82	0.596	01:09:10
Beta-VAE	145.45	2.41	19.18	0.570	01:23:58

Tabla. 5.1: Resultados de reconstrucción con VAE en CelebA.

Los resultados muestran que el modelo **WAE-MMD**(Figura [5.2](#)) es el que presenta un desempeño superior en todas las métricas, seguido de InfoVAE(Figura [5.3](#)) y, finalmente, Beta-VAE(Figura [5.4](#)). Estas diferencias están directamente relacionadas con los principios arquitectónicos de cada modelo, lo cual se refleja de forma significativa en las imágenes reconstruidas.

5.2.2. Análisis visual

La Figura [5.1](#) muestra ejemplos de imágenes reales del conjunto CelebA, utilizadas como referencia. A su vez, en la Figura [5.2](#) se presentan las correspondientes reconstrucciones obtenidas con el modelo WAE-MMD, que como se ha visto, proporciona los mejores resultados.



Figura 5.1: Imágenes reales del conjunto CelebA utilizadas como referencia.

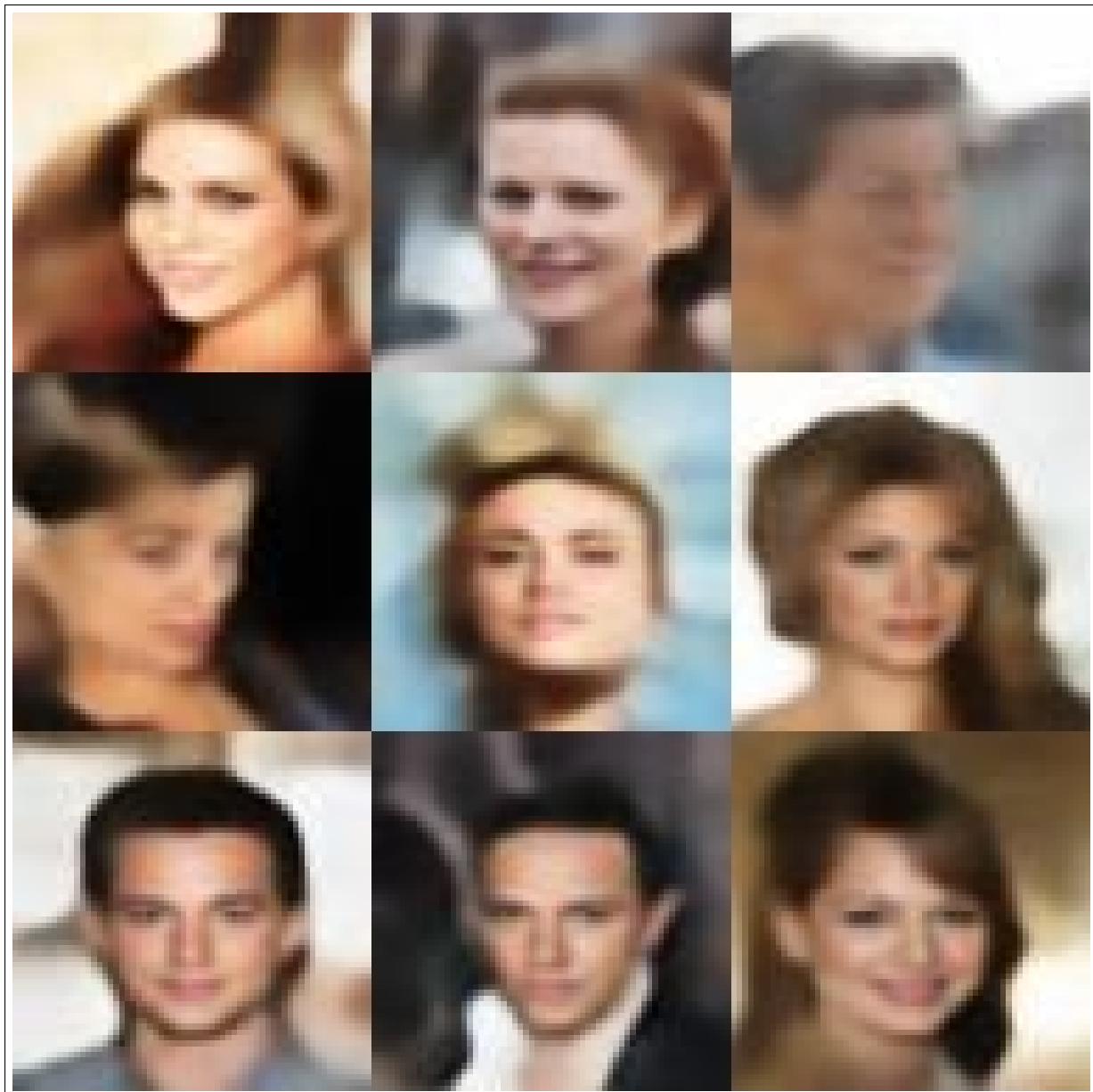


Figura 5.2: Reconstrucciones generadas por el modelo WAE-MMD en CelebA.



Figura 5.3: Reconstrucciones generadas por el modelo Info-vae en CelebA.

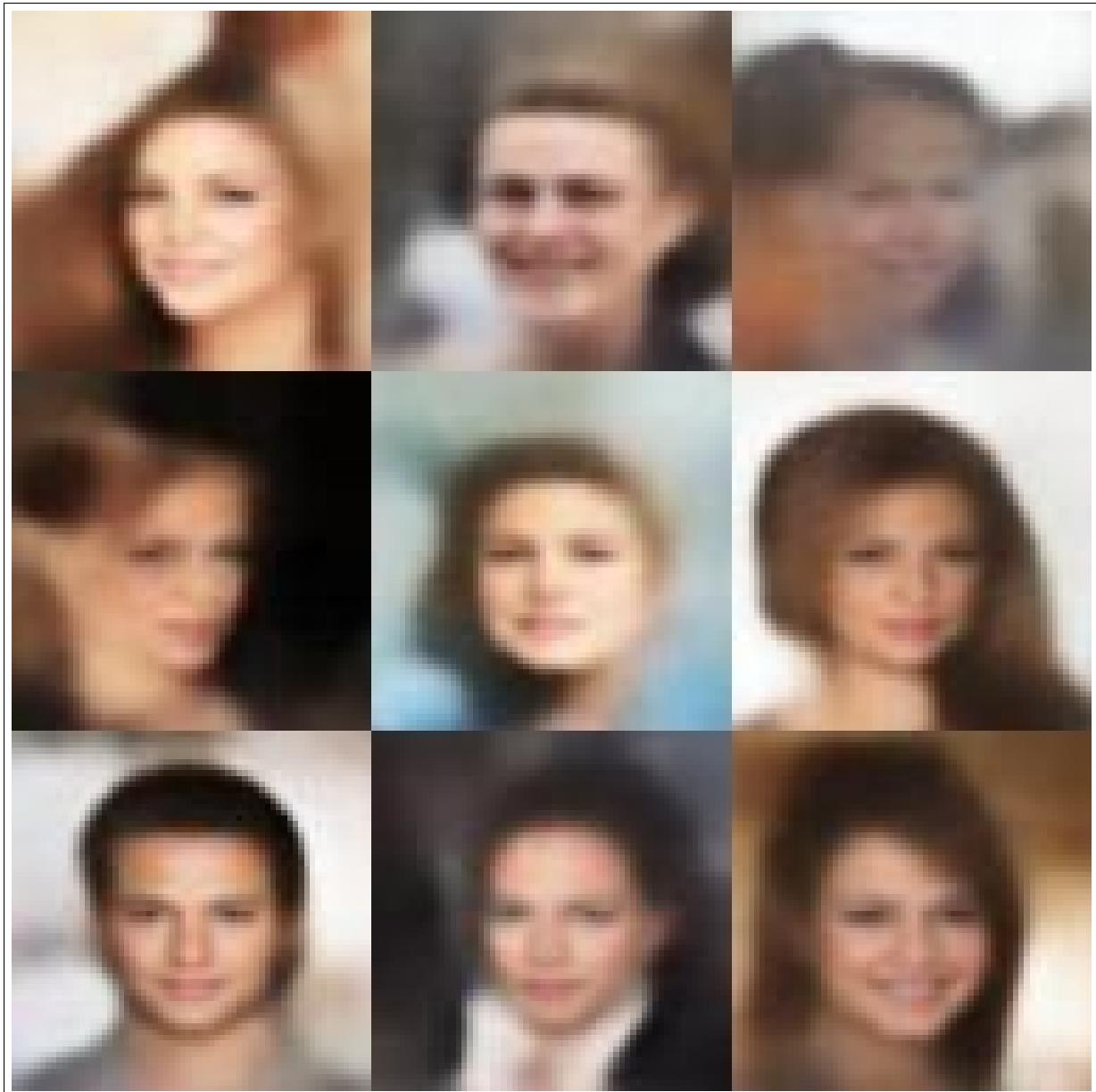


Figura 5.4: Reconstrucciones generadas por el modelo β -vae en CelebA.

Se aprecia que las imágenes generadas por WAE-MMD preservan adecuadamente la morfología facial, especialmente en la disposición de ojos, nariz y boca. Las texturas de piel aparecen suavizadas, pero sin pérdida sustancial de forma, y el fondo es consistente aunque menos detallado. Este comportamiento se explica por el uso de la divergencia MMD en lugar del término KL tradicional, lo que otorga al modelo una mayor libertad para capturar distribuciones más complejas sin forzarlas a una forma gaussiana estricta.

En contraste, los modelos **InfoVAE** y **Beta-VAE**, cuyos resultados no se muestran visualmente en esta sección por razones de brevedad, presentan limitaciones apreciables. En particular, **Beta-VAE**, al utilizar un término de regularización aumentado, genera imágenes más difusas, con rostros menos definidos y pérdidas de información facial crítica. Este fenómeno se manifiesta típicamente en el aplanamiento de contornos, desvanecimiento de la expresión y homogeneidad del fondo. Dichos efectos pueden entenderse como consecuencia directa del compromiso al que fuerza el modelo: aprender representaciones latentes muy disentangladas a costa de sacrificar fidelidad visual.

InfoVAE, por su parte, intenta balancear la entropía latente y la precisión reconstructiva, pero en la práctica esto se traduce en imágenes que, si bien conservan cierta estructura general, tienden a ser más homogéneas y con detalles borrosos en regiones de alta variabilidad (pelo, sonrisa, brillo ocular).

5.2.3. Conclusión parcial

En el contexto de reconstrucción de imágenes faciales en CelebA, el modelo **WAE-MMD** destaca como la opción más equilibrada, ofreciendo una excelente calidad visual a la vez que preserva métricas cuantitativas competitivas. La elección del tipo de regularización y su intensidad impacta directamente en el resultado visual: regularizaciones más suaves (como MMD) permiten conservar detalles faciales críticos, mientras que penalizaciones más estrictas (como en Beta-VAE) conducen a resultados más genéricos y estilizados.

Este análisis deja entrever la importancia de adaptar la formulación del modelo a la complejidad del dominio y al objetivo primario: fidelidad visual frente a compresión semántica. En el siguiente apartado se evaluará cómo estas mismas arquitecturas se comportan ante un conjunto más estructuralmente complejo y menos homogéneo como es el de los UAV.

5.3. Reconstrucción en UAV con modelos VAE

Tras el análisis en CelebA, se extiende la evaluación de los modelos variacionales al dominio de UAVs, un conjunto de datos caracterizado por su heterogeneidad morfológica, mayor complejidad estructural y variabilidad en fondos, escalas y perspectivas. A diferencia de CelebA, donde la estructura facial es relativamente uniforme, las imágenes de drones presentan siluetas irregulares y componentes como hélices, patas, cámaras y brazos, que suponen un reto adicional para los modelos auto-codificadores.

5.3.1. Evaluación cuantitativa

La [Tabla 5.2](#) muestra los resultados obtenidos por WAE-MMD, Beta-VAE e InfoVAE al reconstruir imágenes de UAV. Se observa una degradación general en todas las métricas respecto al experimento previo con CelebA, lo que refleja las mayores dificultades del dominio.

Modelo	FID ↓	IS ↑	PSNR ↑	SSIM ↑	Tiempo H:M:S
WAE-MMD	222.75	3.35	21.49	0.725	00:17:18
Beta-VAE	303.54	2.67	18.63	0.655	00:14:37
InfoVAE	303.79	2.80	19.94	0.680	00:17:31

Tabla. 5.2: Resultados de reconstrucción con VAE en el conjunto UAV.

WAE-MMD([Figura 5.6](#)) vuelve a destacar en todas las métricas, con una mejora notable en SSIM (0.725) pese al incremento general de FID (222.75). Por el contrario, tanto InfoVAE([Figura 5.7](#)) como Beta-VAE([Figura 5.8](#)) obtienen valores superiores a 300 en FID, lo que indica una desviación considerable respecto a la distribución de imágenes reales. La caída en PSNR y SSIM también es significativa, reflejando pérdidas importantes de estructura e intensidad.

5.3.2. Análisis visual

La [Figura 5.5](#) muestra imágenes reales del conjunto UAV. En contraposición, la [Figura 5.6](#) presenta reconstrucciones generadas mediante el modelo WAE-MMD, elegido por su mejor desempeño métrico.

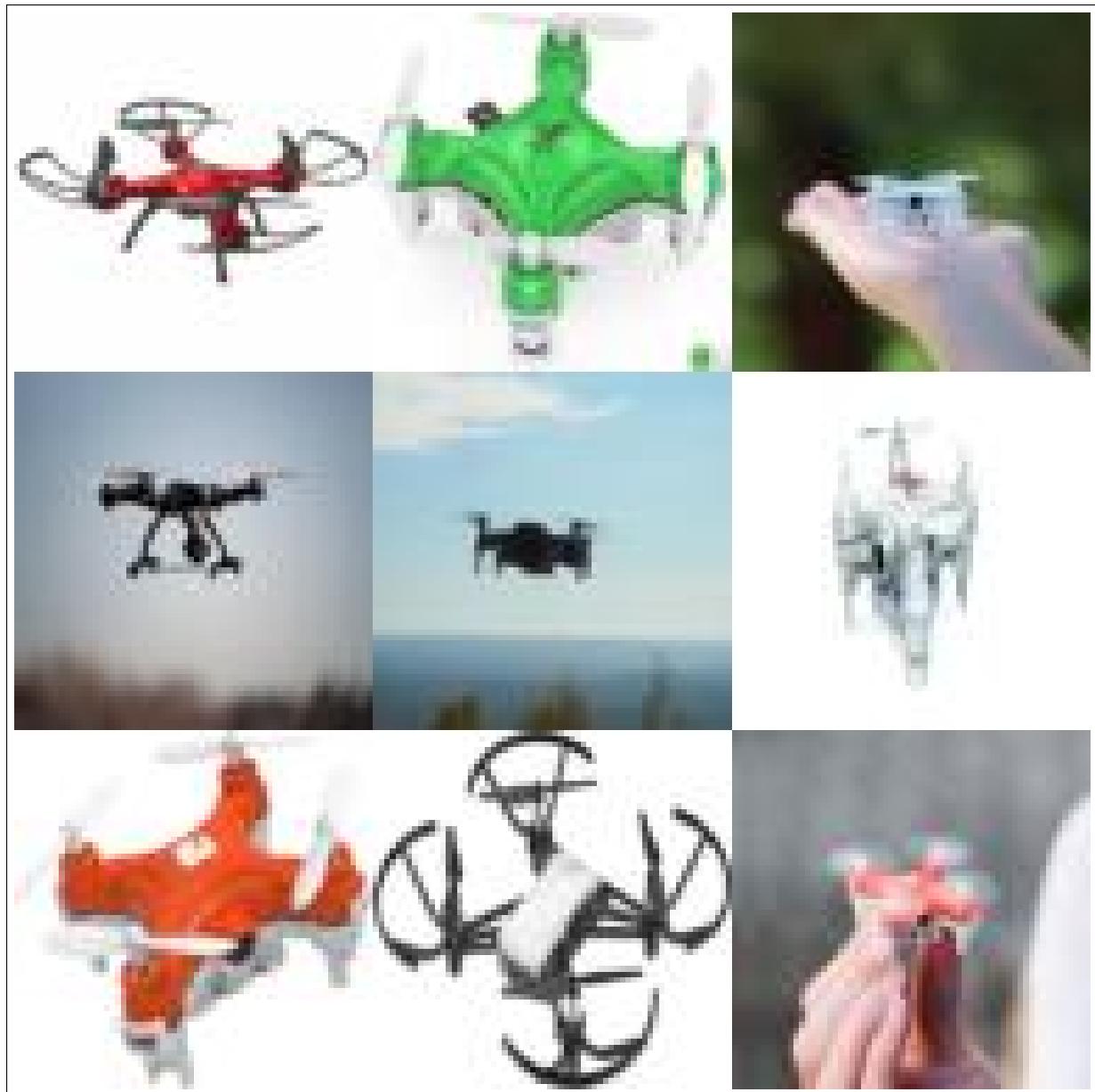


Figura 5.5: Imágenes reales del conjunto UAV.



Figura 5.6: Reconstrucciones generadas por el modelo WAE-MMD en UAV.

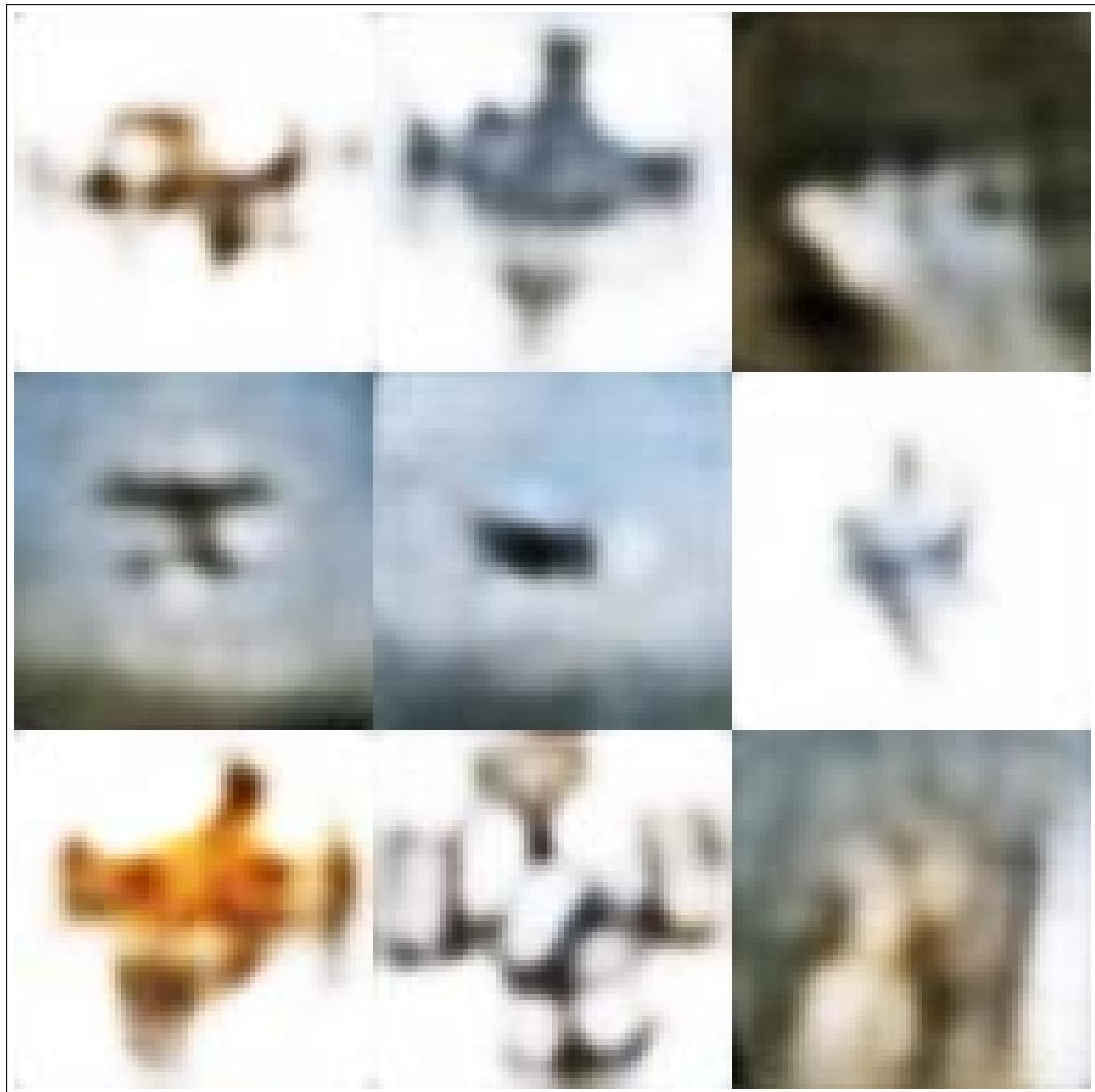


Figura 5.7: Reconstrucciones generadas por el modelo WAE-MMD en UAV.

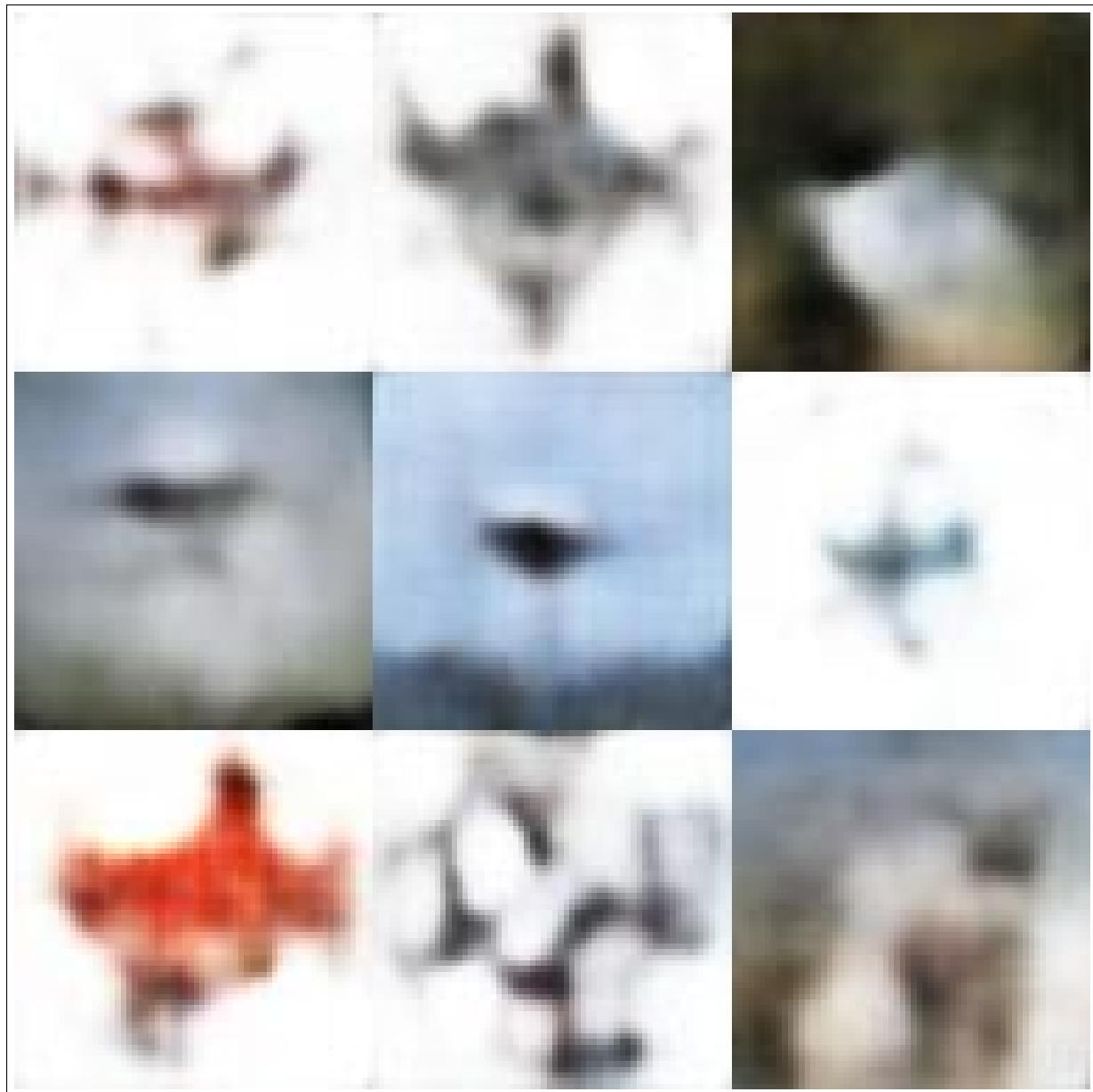


Figura 5.8: Reconstrucciones generadas por el mode lo β -vae en UAV.

A diferencia del dominio facial, donde los modelos pueden apoyarse en una estructura relativamente estable, en UAV se observa que incluso el mejor modelo —WAE-MMD— presenta distorsiones notables. Las formas generales del dron tienden a mantenerse (especialmente la silueta y la orientación general del cuerpo), pero se difuminan elementos importantes como hélices, brazos y patas. El fondo se homogeniza en exceso y aparecen zonas de ruido o confusión entre objeto y entorno. No obstante, las proporciones globales y la orientación espacial se respetan en buena medida, lo que justifica los valores intermedios de SSIM y PSNR.

En el caso de **Beta-VAE**, los efectos de la regularización intensa se acentúan aún más: la reconstrucción resulta visualmente ambigua, con drones que pierden la geometría elemental o que parecen estar fundidos con el fondo. Este desenfoque estructural refleja un latente excesivamente forzado a separabilidad, en detrimento de la precisión visual.

Por su parte, **InfoVAE** muestra un comportamiento mixto: aunque mejora ligeramente en IS respecto a Beta-VAE, la reconstrucción sigue presentando deficiencias significativas en zonas de alta complejidad. Se observan drones sin hélices o con cuerpos incompletos, y regiones desenfocadas que impiden reconocer el tipo de dron original.

5.3.3. Análisis de resultados

En tareas de reconstrucción sobre el conjunto UAV, las arquitecturas VAE revelan sus limitaciones frente a dominios con morfología variable y sin estructuras semánticas tan claras como los rostros. WAE-MMD vuelve a posicionarse como el modelo más robusto, especialmente por su capacidad para conservar relaciones estructurales entre partes del objeto. Sin embargo, los errores visuales observados demuestran que las técnicas actuales basadas en codificación latente aún tienen dificultades notables para representar objetos con geometría explícita.

Este análisis destaca la necesidad de arquitecturas que integren inductores geométricos o mecanismos de atención espacial si se quiere mejorar la reconstrucción de entidades técnicas como los UAV. En la siguiente sección se explorará la capacidad generativa de estos modelos —sin imagen de entrada— y cómo se comportan en entornos aún más exigentes desde cero.

5.4. Generación en CelebA con GANs y modelos de difusión

Una vez evaluadas las capacidades reconstructivas, se analizan ahora los resultados obtenidos al generar imágenes de forma completamente autónoma a partir de ruido latente, sin referencia directa. Esta tarea permite valorar el potencial creativo y expresivo de los modelos generativos, así como su habilidad para aprender la distribución subyacente del conjunto de datos. Para ello se han empleado tanto arquitecturas basadas en redes adversarias (GANs) como modelos de difusión (DDPM).

5.4.1. Evaluación cuantitativa

En la [Tabla 5.3](#) se resumen los resultados de los distintos modelos evaluados en la tarea de generación pura sobre el conjunto CelebA. Las métricas consideradas son FID, IS (con desviación estándar), y el tiempo de entrenamiento.

Modelo	FID ↓	IS ↑ (\pm std)	Tiempo D:H:M:S
DCGAN	74.94	2.81 ± 0.10	0-01:05:17
StyleGAN2-ADA	85.41	3.17 ± 0.08	1-22:44:16
WGAN	91.03	2.58 ± 0.10	0-03:13:32
FDM-EDM	98.22	2.69 ± 0.02	0-06:47:23
WGAN-GP	173.74	2.82 ± 0.12	0-05:20:30

Tabla. 5.3: Resultados de generación en el conjunto CelebA.

Los resultados reflejan un equilibrio delicado entre fidelidad estadística (FID) y diversidad/confianza (IS). Aunque el menor FID corresponde a DCGAN (74.94), el modelo que obtiene el mayor IS es StyleGAN2-ADA (3.17), lo que sugiere que las imágenes generadas por este último son más diversas y perceptualmente consistentes, a pesar de una mayor distancia en el espacio de características.

5.4.2. Análisis visual

Para ilustrar estos resultados, se presentan imágenes de referencia reales del conjunto CelebA en la [Figura 5.1](#). A continuación, se muestran ejemplos generados por StyleGAN2-ADA en la [Figura 5.10](#) y por FDM-EDM en la [Figura 5.11](#).



Figura 5.9: Imágenes reales del conjunto CelebA.



Figura 5.10: Imágenes generadas por StyleGAN2-ADA en CelebA.



Figura 5.11: Imágenes generadas por FDM-EDM en CelebA.

5.4.3. Relación arquitectura–imagen

DCGAN genera imágenes con buena coherencia general y estructuras reconocibles, lo que se alinea con su bajo FID. Sin embargo, la nitidez y realismo facial son inferiores a modelos más avanzados. Aparecen bordes duros, ojos poco definidos y problemas de simetría facial, especialmente en los extremos del conjunto.

WGAN y **WGAN-GP**, aunque teóricamente más estables, muestran FIDs más altos. Esto se refleja en las imágenes como falta de detalles finos, expresiones faciales imprecisas y texturas planas. WGAN-GP, pese a tener un IS relativamente alto (2.82), presenta resultados inconsistentes a nivel visual, con artefactos y asimetrías notorias.

StyleGAN2-ADA es el modelo más sofisticado de los evaluados en este bloque. Su arquitectura basada en capas progresivas y normalización adaptativa permite generar rostros con mayor variedad, simetría y expresión natural. Los ojos, pelo y piel aparecen más detallados y coherentes. Esto explica su IS superior, aunque el FID algo más alto sugiere una distribución de características más alejada de la referencia real, posiblemente por un sesgo hacia rostros idealizados.

FDM-EDM, como modelo de difusión, genera imágenes con contornos suaves y cierta coherencia estructural, pero sufre en nitidez general. Las caras aparecen algo homogéneas, con menos expresividad. Este comportamiento se explica por la naturaleza estocástica del proceso de difusión inversa, que introduce cierto desenfoque si no se ajustan los parámetros de sampling con precisión.

5.4.4. Análisis de resultados

La tarea de generación pura en CelebA revela que las GANs modernas, en particular StyleGAN2-ADA, aún superan a los modelos de difusión en términos de percepción humana, diversidad semántica y nitidez visual. No obstante, la distancia FID no siempre refleja esta ventaja, lo que evidencia una desconexión entre métricas objetivas y apreciación perceptual. Por otro lado, los modelos más simples como DCGAN ofrecen buenos FID pero limitan la calidad visual final, mientras que FDM ofrece un nuevo equilibrio entre coherencia estructural y diversidad, aunque con un coste computacional más alto.

En la siguiente sección se analizará la capacidad de estos mismos modelos para generar imágenes realistas en un dominio más complejo: el conjunto UAV.

5.5. Generación en UAV con GANs y modelos de difusión

El conjunto UAV representa un escenario significativamente más desafiante para la generación de imágenes, debido a su menor tamaño, mayor variabilidad estructural y falta de simetría facial. A diferencia de CelebA, donde los modelos pueden apoyarse en patrones morfológicos estables, las imágenes de drones presentan geometrías abiertas, proporciones irregulares y fondos complejos, lo que exige mayor capacidad de generalización por parte de los modelos.

5.5.1. Evaluación cuantitativa

En la [Tabla 5.4](#) se recogen los resultados obtenidos por cada modelo generativo sobre el conjunto UAV. Las métricas muestran diferencias mucho más marcadas que en CelebA, especialmente en FID, lo que evidencia la complejidad del dominio.

Modelo	FID ↓	IS ↑ (\pm std)	Tiempo D:H:M:S
FDM-EDM	16.80	5.18 ± 0.06	5-02:10:47
StyleGAN2-ADA	26.86	5.20 ± 0.27	7-04:58:32
FDM-VP	41.61	5.60 ± 0.07	2-19:15:56
FDM-EP	90.37	5.09 ± 0.05	2-22:16:34
WGAN-GP	138.71	4.50 ± 0.12	0-03:23:29
DCGAN	183.45	3.93 ± 0.17	0-00:16:10
WGAN	204.72	3.25 ± 0.08	0-01:56:24

Tabla. 5.4: Resultados de generación en el conjunto de drones (UAV).

Destaca claramente el rendimiento de los modelos de difusión, en especial **FDM-EDM**, que obtiene el menor FID (16.80) y un IS competitivo (5.18), indicando no solo buena aproximación estadística al conjunto real, sino también diversidad y coherencia interna. Le sigue **StyleGAN2-ADA** con un IS ligeramente superior (5.20), pero con FID notablemente mayor (26.86), lo que sugiere imágenes más variadas pero algo menos fieles al dominio real.

5.5.2. Análisis visual

La Figura [5.5](#) muestra el dominio UAV, y las Figuras [5.13](#) y [5.16](#), imágenes generadas por GAN y modelos de difusión.

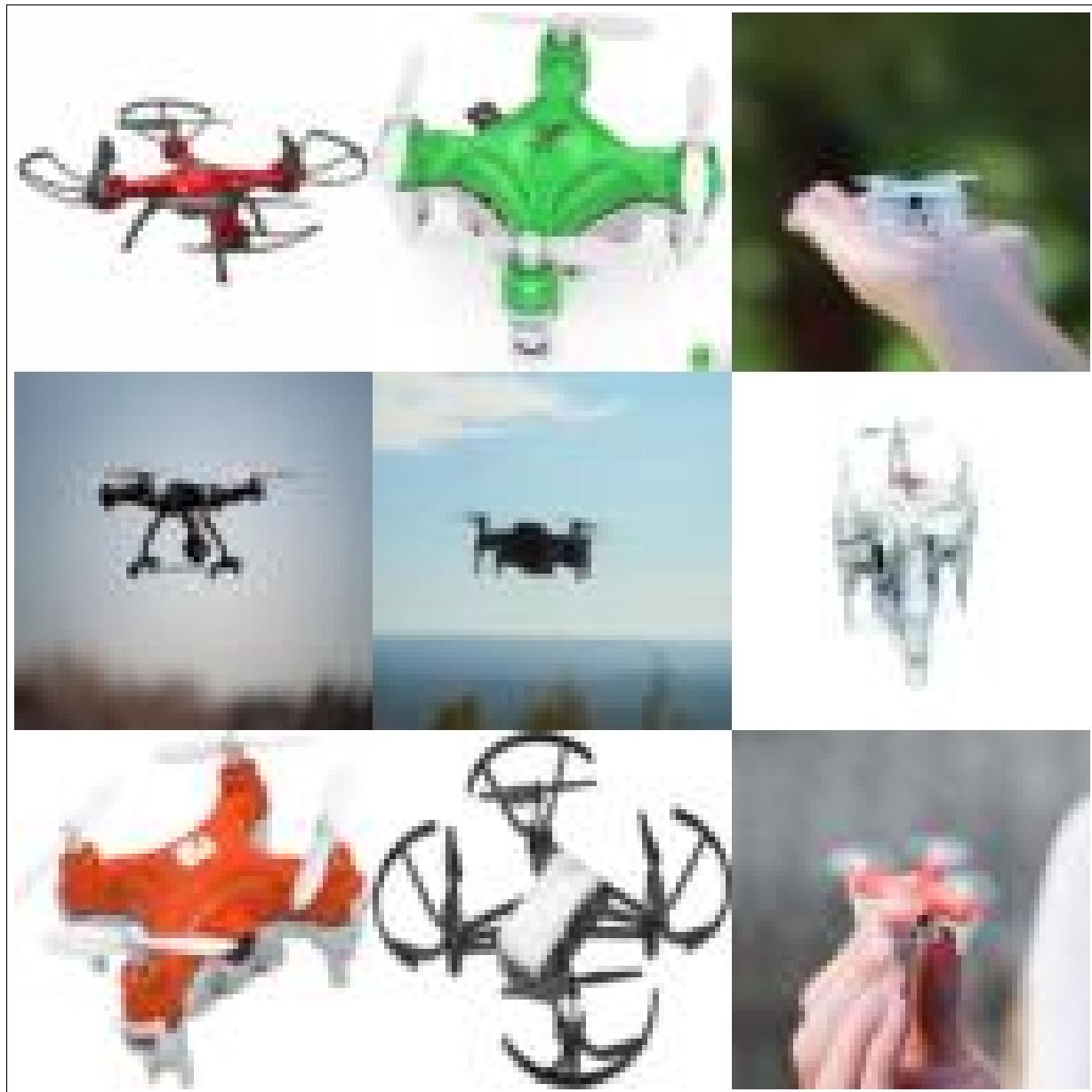


Figura 5.12: Imágenes reales del conjunto UAV.



Figura 5.13: Imágenes generadas por Stylegan-2 en UAV.

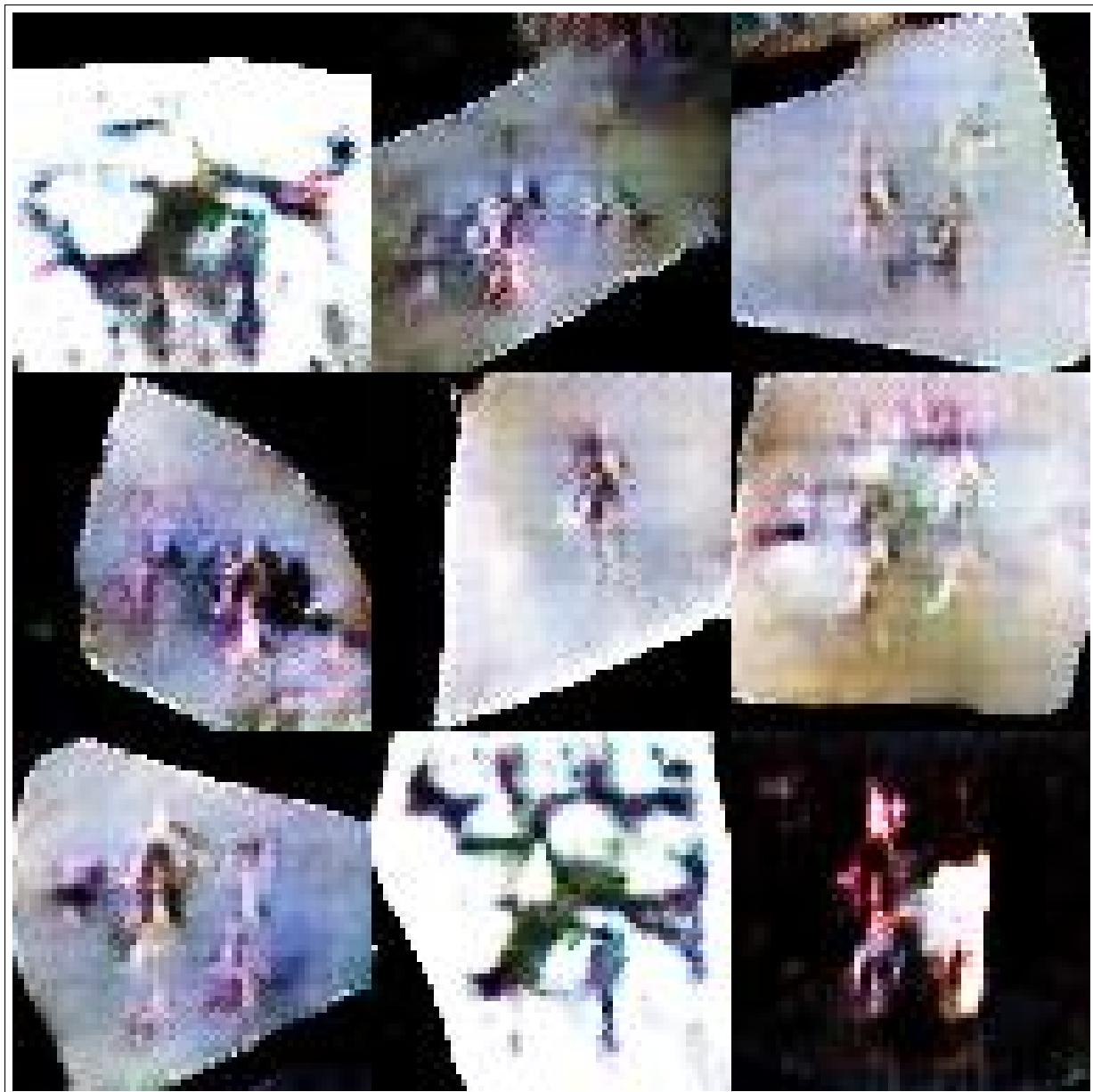


Figura 5.14: Imágenes generadas por DCGAN en UAV.



Figura 5.15: Imágenes generadas por WGAN en UAV.



Figura 5.16: Imágenes generadas por FDM-EDM en UAV.

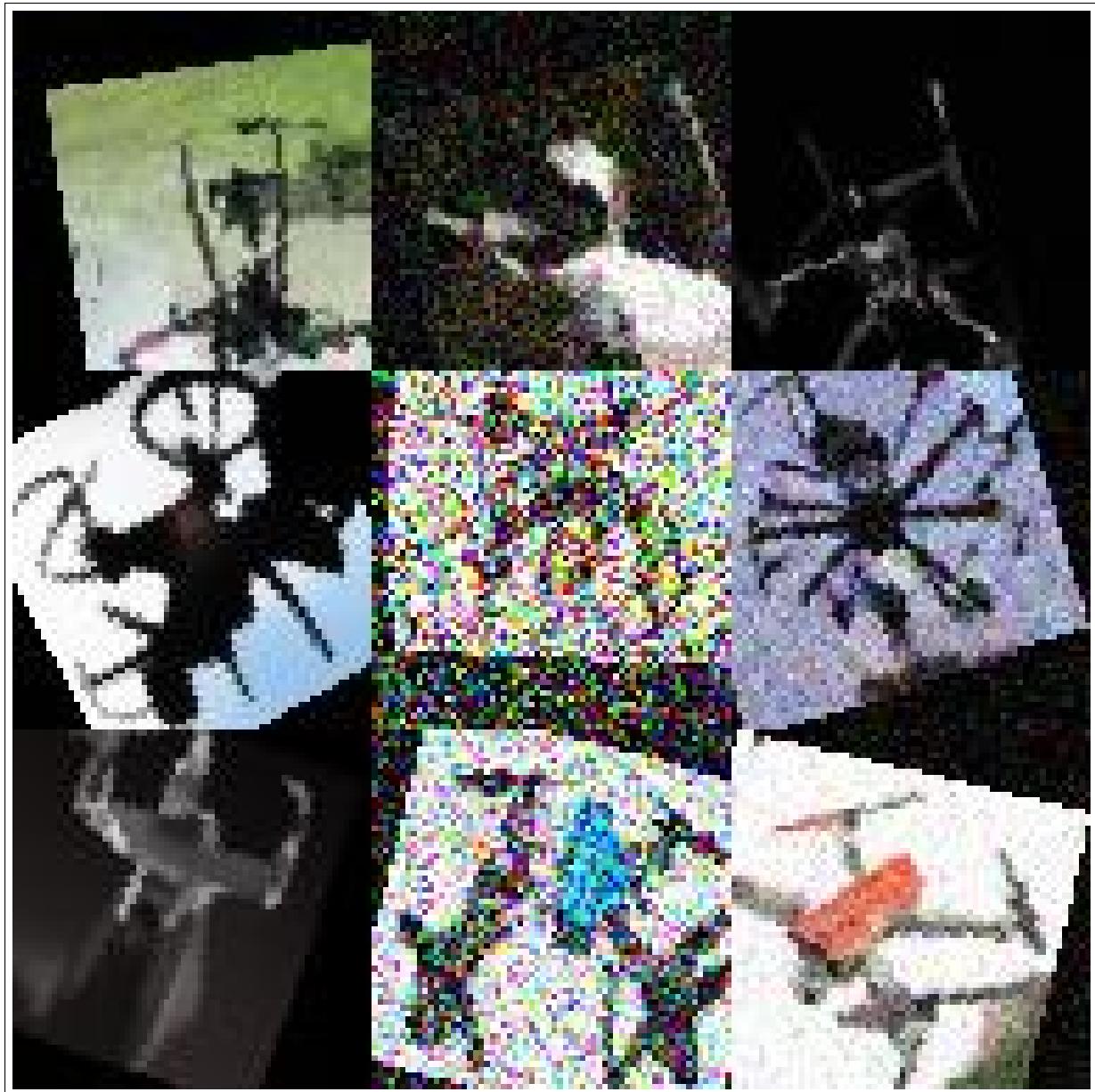


Figura 5.17: Imágenes generadas por FDM-VP en UAV.

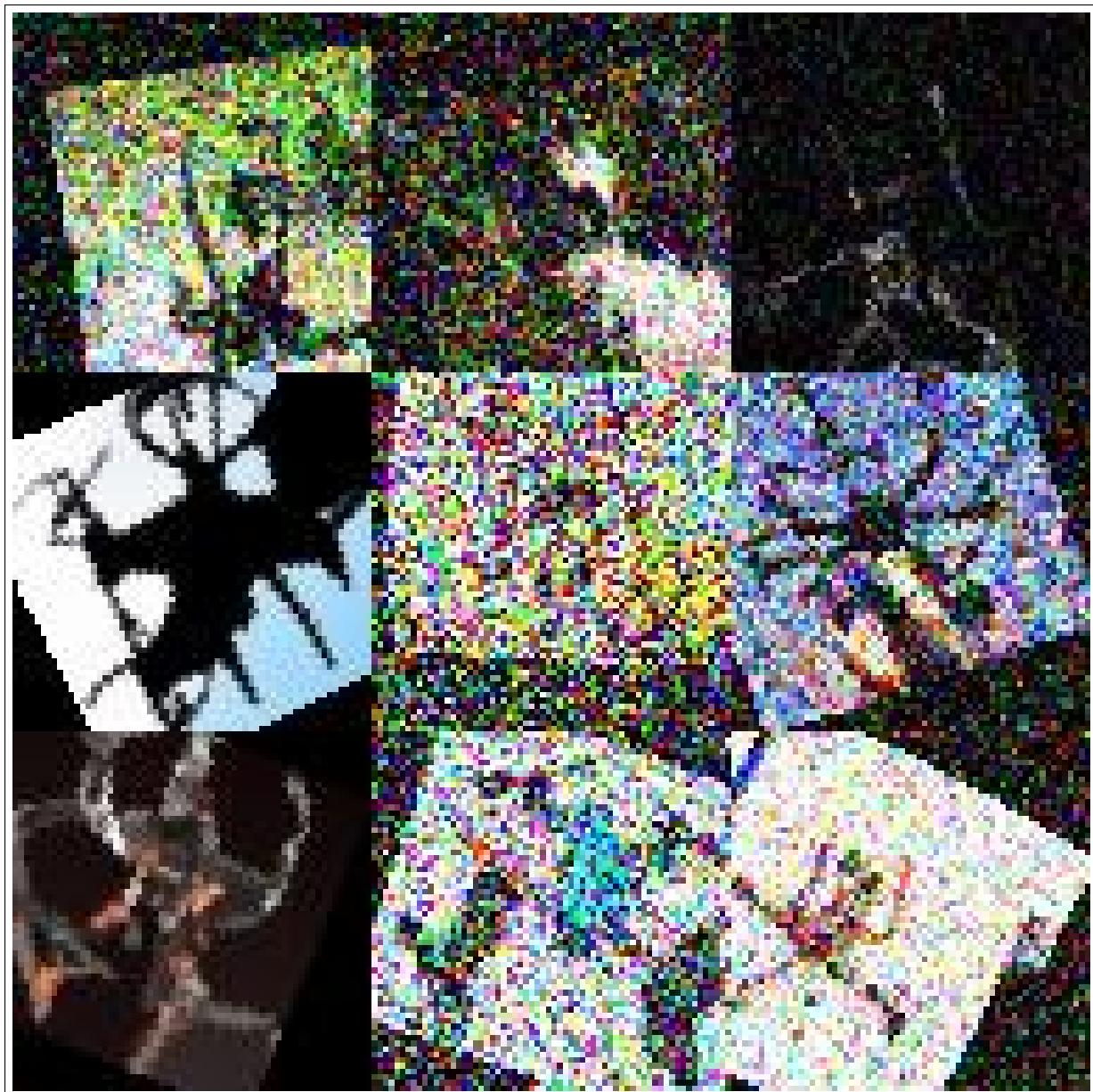


Figura 5.18: Imágenes generadas por FDM-EP en UAV.

5.5.3. Relación arquitectura–imagen

FDM-EDM produce imágenes sorprendentemente definidas, con formas de drones reconocibles, simetría estructural, y diferenciación clara entre el objeto y el fondo. Las hélices, cuerpos y brazos aparecen proporcionados y situados con lógica espacial. Este comportamiento está directamente relacionado con el uso de trayectorias de denoising estables, que permiten modelar la distribución de datos con alta precisión espacial y semántica.

FDM-VP genera también resultados de alta calidad, aunque en ocasiones tiende a suavizar demasiado las texturas, haciendo que algunas partes del dron se integren excesivamente con el fondo. Esto puede deberse a la naturaleza más difusa de su proceso de sampling. No obstante, sus métricas IS (5.60) y FID (41.61) reflejan un compromiso favorable entre diversidad y fidelidad.

FDM-VE se comporta de forma intermedia respecto a los dos anteriores. Aunque logra mantener cierta estructura general en los drones, muestra mayor tendencia a introducir ruido de alta frecuencia en el fondo, así como inconsistencias en los contornos. Este comportamiento se explica por la agresividad del escalado de varianza característico del esquema VE, que puede amplificar imperfecciones durante el sampling. Aun así, su inclusión en el estudio resulta relevante como referencia del comportamiento bajo trayectorias explosivas.

StyleGAN2-ADA, aunque queda por detrás en FID, logra una alta variabilidad y riqueza morfológica en las muestras. Sus resultados destacan por su realismo estético y composición visual, pero a veces sacrifican precisión estructural: es común observar drones con configuraciones atípicas o con partes deformadas. Esta observación sugiere que el modelo aprende una representación estilizada del dominio, más que una reconstrucción estricta.

DCGAN y **WGAN** obtienen los peores resultados tanto en FID como IS. Sus imágenes (Figura 5.155.14) muestran estructuras irreconocibles, artefactos geométricos, y fusiones erróneas entre fondo y objeto. El bajo rendimiento se atribuye a la falta de mecanismos de normalización y a la limitada expresividad de las capas convolucionales tradicionales, incapaces de capturar relaciones espaciales complejas.

WGAN-GP mejora ligeramente, pero aún presenta inconsistencias graves. Aunque su IS es más alto que DCGAN (4.50 frente a 3.93), las muestras muestran morfologías incoherentes, con drones incompletos o visualmente deformes.

5.5.4. Análisis de resultados

Los resultados en UAV consolidan la ventaja de los modelos de difusión frente a las GANs clásicas en tareas de generación estructural. FDM-EDM, en particular, demuestra una capacidad notable para preservar coherencia espacial, simetría y detalle incluso en dominios sin referencias morfológicas estándar. StyleGAN2-ADA ofrece una alternativa potente, especialmente en términos de variedad y estética, aunque con más margen de error estructural.

Las GANs tradicionales, en cambio, se ven ampliamente superadas en este dominio, lo que sugiere que la capacidad de representar estructuras técnicas requiere modelos que integren procesos más controlados de reconstrucción paso a paso, como los que permiten los métodos de difusión.

En la siguiente sección se presentará una visión comparativa más general, integrando los resultados obtenidos en todos los conjuntos y modelos para extraer conclusiones globales sobre la relación arquitectura–dominio–calidad visual.

5.6. Discusión comparativa y síntesis

Con el objetivo de integrar y contrastar los resultados obtenidos a lo largo de este estudio, esta sección presenta una discusión comparativa entre los diferentes modelos generativos evaluados, considerando tanto el tipo de arquitectura como la tarea (reconstrucción o generación) y el dominio visual (rostros humanos en CelebA y vehículos UAV).

En la [Tabla 5.5](#) se recoge de forma sintética el desempeño cuantitativo de cada modelo en los distintos escenarios experimentales. La tabla organiza los datos según cuatro dimensiones clave: el modelo utilizado, su tipo (VAE, GAN o FDM), el conjunto de datos sobre el que se ha evaluado y la tarea específica (reconstrucción o generación). Las métricas reflejan la fidelidad estadística (FID), la diversidad y confianza perceptual (IS), y en los casos donde procede, las métricas de reconstrucción por píxel (PSNR y SSIM).

Modelo	Tipo	Conjunto	Tarea	FID ↓	IS ↑	PSNR ↑	SSIM ↑
WAE-MMD	VAE	CelebA	Reconstrucción	122.90	2.65	21.79	0.621
InfoVAE	VAE	CelebA	Reconstrucción	139.07	2.51	20.82	0.596
Beta-VAE	VAE	CelebA	Reconstrucción	145.45	2.41	19.18	0.570
WAE-MMD	VAE	UAV	Reconstrucción	222.75	3.35	21.49	0.725
InfoVAE	VAE	UAV	Reconstrucción	303.79	2.80	19.94	0.680
Beta-VAE	VAE	UAV	Reconstrucción	303.54	2.67	18.63	0.655
DCGAN	GAN	CelebA	Generación	74.94	2.81	—	—
StyleGAN2-ADA	GAN	CelebA	Generación	85.41	3.17	—	—
WGAN	GAN	CelebA	Generación	91.03	2.58	—	—
FDM-EDM	FDM	CelebA	Generación	98.22	2.69	—	—
WGAN-GP	GAN	CelebA	Generación	173.74	2.82	—	—
FDM-EDM	FDM	UAV	Generación	16.80	5.18	—	—
StyleGAN2-ADA	GAN	UAV	Generación	26.86	5.20	—	—
FDM-VP	FDM	UAV	Generación	41.61	5.60	—	—
FDM-EP	FDM	UAV	Generación	90.37	5.09	—	—
WGAN-GP	GAN	UAV	Generación	138.71	4.50	—	—
DCGAN	GAN	UAV	Generación	183.45	3.93	—	—
WGAN	GAN	UAV	Generación	204.72	3.25	—	—

Tabla. 5.5: Comparativa global entre modelos generativos evaluados en CelebA y UAV, diferenciando por arquitectura, tarea y conjunto. Las celdas con “—” indican métricas no aplicables en generación pura.

5.6.1. Arquitectura y rendimiento: VAE, GAN y FDM

Los modelos basados en **autoencoders variacionales (VAE)** muestran un comportamiento notablemente dependiente del tipo de regularización que aplican. En ambos conjuntos (CelebA y UAV), el modelo **WAE-MMD** obtiene los mejores resultados, tanto cuantitativamente (mejores FID, IS, PSNR y SSIM) como visualmente (véase Figuras 5.2 y 5.6). Esta ventaja se atribuye al uso de la divergencia MMD, que permite preservar más variabilidad y detalle semántico en el espacio latente.

En contraposición, **Beta-VAE** y **InfoVAE** muestran degradación progresiva de la calidad visual a medida que aumenta la presión de disentanglement. Esto es especialmente notorio en UAV, donde los modelos sufren para reconstruir estructuras irregulares y detalles como hélices o soportes. Las métricas PSNR y SSIM en UAV son, en todos los casos, más bajas que en CelebA, reflejando la mayor complejidad estructural del dominio técnico frente al facial.

En el caso de los modelos de **generación pura**, las **GANs tradicionales** como DCGAN y WGAN funcionan razonablemente bien en CelebA, pero colapsan en UAV. Las imágenes generadas por estos modelos en UAV presentan artefactos, formas irreconocibles y falta de coherencia estructural (ver Figura 5.15), lo que se traduce

en FID extremadamente altos (>180) y IS por debajo de 4.

StyleGAN2-ADA, en cambio, se adapta mejor a ambos dominios. Aunque no alcanza el mejor FID, su IS es el más alto en ambos conjuntos (3.17 en CelebA, 5.20 en UAV), y visualmente genera imágenes detalladas, diversas y perceptualmente agradables. Su capacidad de mantener consistencia en la variación semántica —a través de su arquitectura progresiva y su regularización adaptativa— la convierte en una solución robusta y visualmente potente.

Finalmente, los **modelos de difusión (FDM)** se posicionan como la arquitectura más eficaz en dominios complejos. **FDM-EDM** obtiene el mejor FID global (16.80) en UAV, y genera imágenes notablemente precisas en términos estructurales, con buena definición de contorno, orientación y simetría (Figura 5.17). Aunque su IS es algo inferior al de StyleGAN2-ADA, esto puede atribuirse a la menor aleatoriedad perceptual, lo que también explica su menor varianza.

5.6.2. Tarea: reconstrucción vs. generación

Las tareas de **reconstrucción** presentan un escenario más estable, con menos exigencias creativas, lo que favorece a los modelos VAE. Sin embargo, incluso en este entorno, las diferencias de arquitectura tienen un impacto directo en la calidad visual obtenida. Las imágenes reconstruidas por WAE-MMD preservan detalles esenciales, mientras que las de Beta-VAE resultan difusas y poco informativas.

En la tarea de **generación**, las diferencias entre arquitecturas se amplifican. Mientras que las GANs sufren en dominios con alta variabilidad geométrica (como UAV), los modelos de difusión sobresalen al mantener consistencia estructural incluso en ausencia de patrones morfológicos repetidos.

5.6.3. Dominio: CelebA vs. UAV

El dominio de **rostros humanos** representa un entorno más "amigable" para los modelos generativos. La repetitividad de las formas, la simetría facial y la homogeneidad del fondo permiten que incluso arquitecturas simples generen imágenes plausibles. Esto se refleja en los bajos FID de DCGAN (74.94) o WGAN (91.03) en CelebA, frente a sus pésimos resultados en UAV (>180).

El dominio **UAV**, en cambio, pone a prueba la verdadera capacidad de modelado

estructural. Solo modelos avanzados como StyleGAN2-ADA o FDM-EDM logran resultados competitivos, con diferencias notables tanto en métricas como en percepción visual. Esto sugiere que la capacidad de generalización de un modelo depende no solo de su arquitectura, sino también de su adecuación al tipo de datos.



Figura 5.19: Comparativa visual de resultados generativos en UAV: imagen real (izquierda), StyleGAN2-ADA, FDM-EDM y WAE-MMD (reconstrucción). Se aprecian diferencias en textura, definición y coherencia estructural.

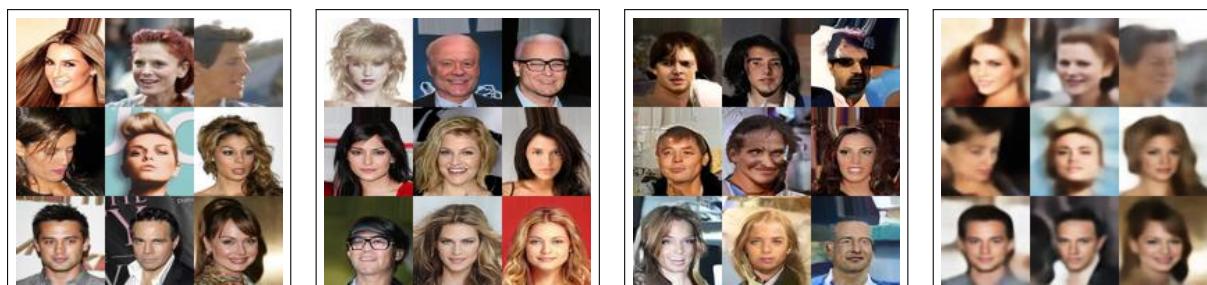


Figura 5.20: Comparativa visual de resultados generativos en Celeb-A: imagen real (izquierda), StyleGAN2-ADA, FDM-EDM y WAE-MMD (reconstrucción). Se aprecian diferencias en textura, definición y coherencia estructural.

5.6.4. Reflexión final

El análisis comparativo muestra que:

- **WAE-MMD** es la opción más sólida en reconstrucción, especialmente por su balance entre regularización y expresividad.
- **StyleGAN2-ADA** domina la generación en entornos estilizados o con patrones repetitivos (como rostros), ofreciendo alta diversidad y realismo perceptual.
- **FDM-EDM** sobresale en tareas que requieren modelado geométrico preciso, como la generación de UAVs.
- Las métricas como FID o IS deben interpretarse junto a la observación visual, ya que no siempre capturan aspectos semánticos o morfológicos relevantes.

Este análisis fundamenta la elección de los modelos a futuro no solo en función de sus métricas estándar, sino considerando también el dominio de aplicación y el objetivo visual deseado. El próximo capítulo planteará líneas de mejora y desarrollos futuros que surgen a partir de estas observaciones.

Capítulo 6

Resumen y conclusiones

Este trabajo ha tenido como objetivo principal analizar, comparar y comprender el comportamiento de diferentes arquitecturas generativas aplicadas a la síntesis y reconstrucción de imágenes en dos dominios visuales contrastados: **rostros humanos** (CelebA) y **vehículos aéreos no tripulados** (UAV). A través de una batería de experimentos estructurados, se ha evaluado el desempeño de modelos VAE, GAN y de difusión tanto en tareas de reconstrucción como de generación pura, utilizando métricas objetivas (FID, IS, PSNR, SSIM) y análisis visual sistemático.

6.1. Resumen general del trabajo

Este Trabajo Fin de Grado se enmarca en el ámbito de la inteligencia artificial generativa, con especial énfasis en la generación de imágenes sintéticas aplicadas a dos dominios diferenciados: los rostros humanos —utilizando el conjunto de datos CelebA— y los vehículos aéreos no tripulados (UAVs), representados mediante un dataset propio construido y ampliado específicamente para este trabajo. El objetivo general ha sido doble: por un lado, comprender en profundidad las arquitecturas generativas más relevantes en la literatura; y por otro, evaluar de forma sistemática su aplicabilidad, rendimiento y utilidad práctica en tareas donde los datos reales son escasos o difíciles de obtener.

El trabajo se estructura en torno a tres ejes fundamentales:

- **Revisión y análisis de modelos generativos:** se ha realizado un estudio exhaustivo de tres grandes familias de modelos generativos profundos —Autoencoders

Variacionales (VAE), Redes Generativas Adversariales (GAN) y Modelos de Difusión (DDPM)—, describiendo sus fundamentos teóricos, su evolución histórica, sus variantes principales y su adecuación al problema de síntesis de imágenes.

- **Implementación y experimentación:** se han entrenado múltiples configuraciones de modelos en dos dominios visuales contrastados, siguiendo un pipeline común de preprocesado, entrenamiento, validación y generación. Se han utilizado arquitecturas como DCGAN, WGAN-GP, StyleGAN2-ADA, β -VAE, VQ-VAE, FDM-EDM y FDM-VP, entre otras.
- **Evaluación objetiva y cualitativa:** se han aplicado métricas estandarizadas como FID, IS, PSNR y SSIM para comparar el desempeño de los modelos. Adicionalmente, se han incluido análisis visuales y valoraciones cualitativas para interpretar los resultados y detectar patrones no capturados por las métricas numéricas.

De forma complementaria, el trabajo incorpora un capítulo de antecedentes teóricos que contextualiza el desarrollo técnico dentro de los marcos más amplios de la inteligencia artificial, el aprendizaje automático y la ciencia de datos. Se abordan aspectos como la evolución histórica de la IA, el papel de la generación sintética como técnica de aumentación de datos, los riesgos asociados al sobreajuste a datos artificiales, y las estrategias de transferencia de dominio.

A nivel metodológico, se ha seguido un enfoque reproducible y transparente. Todos los modelos han sido entrenados con el mismo conjunto de condiciones iniciales, en entornos controlados, y los resultados han sido documentados con precisión. Además, el trabajo ha prestado especial atención al equilibrio entre costes computacionales y calidad visual, incluyendo estimaciones detalladas de recursos y tiempo, así como una planificación temporal realista dividida en fases.

En resumen, este TFG ofrece una visión global, crítica y aplicada del estado actual de la generación de imágenes mediante redes neuronales profundas, subrayando tanto sus capacidades como sus limitaciones, y sentando las bases para futuras investigaciones y desarrollos en este campo.

6.2. Lecciones aprendidas

A lo largo de este Trabajo Fin de Grado se ha llevado a cabo un análisis sistemático de las principales técnicas generativas para la síntesis de imágenes, evaluando

tanto su rendimiento cuantitativo como su utilidad práctica en dominios con escasez de datos reales. Las conclusiones más relevantes que se derivan de este estudio son las siguientes:

- **Los modelos generativos actuales son capaces de producir imágenes altamente realistas**, pero su rendimiento depende críticamente del tipo de arquitectura, del tamaño del conjunto de entrenamiento y de las características del dominio. Mientras que las GAN destacan por su fidelidad visual y rapidez, los VAE ofrecen mayor interpretabilidad latente, y los DDPM alcanzan el estado del arte en calidad a costa de un mayor coste computacional.
- **La generación sintética se consolida como una herramienta eficaz de aumentación de datos**, especialmente en escenarios donde la recopilación de imágenes reales resulta costosa, peligrosa o limitada (como sucede con los UAVs). No obstante, el uso indiscriminado de muestras artificiales puede inducir sesgos o sobreajuste si no se acompaña de mecanismos de validación y transferencia de dominio.
- **La transferencia de dominio resulta esencial** cuando se pretende aplicar modelos entrenados sobre datos sintéticos a situaciones reales. La diferencia entre distribuciones —*domain gap*— se traduce en caídas de rendimiento significativas si no se utilizan técnicas como fine-tuning supervisado, adaptación adversarial o normalización estadística.
- **La evaluación objetiva mediante métricas como FID, IS, PSNR y SSIM proporciona un marco cuantitativo robusto**, pero no sustituye a la evaluación cualitativa. La calidad perceptual, la coherencia semántica o la utilidad práctica de las imágenes generadas deben analizarse también desde una perspectiva visual y contextual.
- **La viabilidad técnica y económica de entrenar modelos generativos depende del entorno computacional disponible**. Este trabajo ha mostrado que existen rutas eficientes —mediante GPUs locales o arquitecturas ligeras— que permiten obtener resultados competitivos sin necesidad de grandes infraestructuras como las estaciones DGX o servicios cloud premium.

En conjunto, este trabajo demuestra que el uso combinado de modelos generativos, técnicas de aumentación de datos y estrategias de transferencia de dominio constituye un enfoque sólido y versátil para ampliar datasets, robustecer modelos discriminativos y afrontar tareas visuales complejas en contextos técnicos y operativos reales.

6.3. Líneas futuras de investigación

A partir de los resultados obtenidos y del análisis crítico realizado, se identifican varias líneas de trabajo prometedoras que podrían abordarse en proyectos futuros:

1. **Optimización del proceso de generación:** Aunque los modelos de difusión logran resultados sobresalientes en términos de realismo, su alto coste de inferencia limita su aplicabilidad práctica. Una dirección futura relevante consiste en explorar variantes aceleradas, como DDIM, Consistency Models o modelos distilados, que permiten reducir el número de pasos de sampling sin pérdida sustancial de calidad.
2. **Control semántico más fino:** La generación condicional basada en etiquetas ofrece un control limitado sobre atributos complejos. Integrar enfoques como *ControlNet*, *Textual Inversion* o *Classifier-Free Guidance* podría mejorar el control de estilo, contexto y pose, abriendo nuevas posibilidades para generación dirigida en UAVs con escenarios específicos (por ejemplo, drones en formaciones militares o entornos urbanos).
3. **Entrenamiento multimodal y multirresolución:** La capacidad de trabajar con entradas heterogéneas —texto, imagen, mapa de segmentación, vídeo— permitiría explotar sinergias entre tareas y mejorar la generalización. Asimismo, la generación a resoluciones superiores (256x256 o 512x512) requiere optimizaciones en arquitectura y entrenamiento progresivo, así como técnicas de super-resolución.
4. **Evaluación humana y perceptual:** El trabajo se ha centrado en métricas automatizadas, pero la incorporación de estudios de percepción humana permitiría validar de forma más holística la utilidad y credibilidad de las imágenes generadas, especialmente en aplicaciones donde la percepción subjetiva es crítica (simuladores de entrenamiento, entornos de defensa, realidad virtual).
5. **Integración en pipelines de entrenamiento reales:** Una extensión natural del proyecto sería desplegar los datos generados en tareas de clasificación o detección realistas, midiendo su impacto directo en rendimiento de modelos discriminativos como YOLOv8, Faster R-CNN o Vision Transformers. Esto permitiría validar empíricamente la utilidad de la generación sintética dentro de un flujo completo de machine learning aplicado.
6. **Ánalisis ético y trazabilidad de modelos generativos:** En línea con preocupaciones actuales sobre el uso malicioso de modelos generativos, otra línea

importante es el estudio de mecanismos de detección de imágenes sintéticas, trazabilidad de muestras y evaluación del sesgo inducido por los datasets y arquitecturas utilizadas. El cumplimiento de marcos normativos (como la futura AI Act de la UE) exigirá mayor atención a estos aspectos.

Estas líneas no solo extienden el alcance técnico del presente trabajo, sino que refuerzan su aplicabilidad en entornos industriales, académicos y sociales, contribuyendo al desarrollo de una inteligencia artificial generativa más robusta, controlable y responsable.

6.4. Conclusión Final

Este Trabajo Fin de Grado ha tratado de ofrecer una visión exhaustiva y aplicada del estado actual de la generación sintética de imágenes mediante modelos generativos profundos. A través del análisis teórico, la implementación práctica y la evaluación sistemática, se ha demostrado la utilidad de estas técnicas en un caso de aplicación concreto: la generación de imágenes de UAVs para mejorar sistemas de detección visual.

Más allá de los resultados técnicos, el trabajo ha servido para evidenciar la necesidad de integrar distintas disciplinas —inteligencia artificial, ciencia de datos, visión por computador— bajo un enfoque riguroso, reproducible y éticamente consciente. El potencial transformador de los modelos generativos es incuestionable, pero su aprovechamiento efectivo exige un marco metodológico sólido, validaciones empíricas y una atención constante a sus implicaciones éticas y operativas.

Se espera que este estudio constituya no sólo una contribución académica puntual, sino una base útil para futuras investigaciones, aplicaciones industriales o desarrollos científicos en el área de la IA generativa.

Recursos utilizados

A continuación se recogen los recursos principales empleados durante el desarrollo del presente trabajo, diferenciando entre conjuntos de datos utilizados y repositorios de código que han servido como base o referencia.

Conjuntos de datos

- **UAV Detection Dataset – Kaggle**

<https://www.kaggle.com/datasets/nelyg8002000/uav-detection-dataset-images>

Conjunto de imágenes de drones en distintos entornos y condiciones visuales.

Se ha empleado como base para los experimentos de reconstrucción y generación en el dominio UAV. Se ha preprocesado y reducido a un subconjunto curado manualmente.

- **CelebA – CelebFaces Attributes Dataset**

<https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>

Base de datos de rostros humanos ampliamente utilizada como benchmark en generación de imágenes. Utilizada para entrenar y comparar los modelos en el dominio facial.

Repositorios de código

- **Repositorio principal del TFG (Galaxyraul)**

<https://github.com/Galaxyraul/TFG>

Implementación propia del trabajo final de grado. Incluye los scripts de entrenamiento, inferencia, evaluación y generación de imágenes con múltiples arquitecturas.

- **PyTorch-GAN (Erik Lindernoren)**

<https://github.com/eriklindernoren/PyTorch-GAN>

Repositorio de referencia con implementaciones base de múltiples variantes de GAN. Ha servido como apoyo para la construcción de modelos en el repositorio principal del TFG.

- **PyTorch-VAE (AntixK)**

<https://github.com/AntixK/PyTorch-VAE>

Colección de variantes de autoencoders variacionales en PyTorch. Se ha utilizado como punto de partida para el diseño y adaptación de los modelos VAE empleados.

- **StyleGAN2-ADA (NVIDIA)**

<https://github.com/NVlabs/stylegan2-ada-pytorch>

Repositorio oficial de StyleGAN2 con data augmentation adaptativo. Se ha empleado para generar imágenes faciales y de UAV tras su entrenamiento con conjuntos reducidos.

- **FDM (Fast Diffusion Models – SAIL-SG)**

<https://github.com/sail-sg/FDM>

Implementación eficiente de modelos de difusión. Ha sido utilizada para experimentar con FDM-EDM y FDM-VP, especialmente en la generación de imágenes técnicas.

Bibliografía

- [1] Wolfram. Building a data science pipeline. <https://www.youtube.com/watch?v=cugl5t-W1sE>, July 2020. URL <https://www.youtube.com/watch?v=cugl5t-W1sE>. YouTube video.
- [2] databacc. The data science venn diagram - data science: An introduction - 2.2. <https://www.youtube.com/watch?v=r2I3IDKwyMw>. URL [https://www.youtube.com/watch?v=\[AQUÃD_EL_ID_DEL_VIDEO\]](https://www.youtube.com/watch?v=[AQUÃD_EL_ID_DEL_VIDEO]). YouTube video.
- [3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [5] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.
- [7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- [8] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [12] Liang Zhao, Zhenwei Tang, and Yimin Wang. Drone detection and tracking based on deep learning: A survey. *Sensors*, 20(10):2836, 2020.
- [13] Robert J. Sternberg. *Beyond IQ: A Triarchic Theory of Human Intelligence*. Cambridge University Press, 1985.
- [14] Howard Gardner. *Frames of Mind: The Theory of Multiple Intelligences*. Basic Books, 1983.
- [15] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3rd edition, 2016.
- [16] Alan M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 1950. doi: 10.1093/mind/LIX.236.433.
- [17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 779–788, 2016.
- [18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, pages 234–241. Springer, 2015.
- [19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [20] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–54, 1996.
- [21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, and et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.

- [22] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- [23] Patrice Y Simard, Daniel Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, pages 958–962, 2003.
- [24] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *CVPR*, 2019.
- [25] Yunhao Chen, Zihui Yan, and Yunjie Zhu. A comprehensive survey for generative data augmentation. *Neurocomputing*, page 128167, 2024.
- [26] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7167–7176, 2017.
- [27] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.
- [28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [29] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [30] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [31] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3722–3731, 2017.
- [32] Han Zhao, Shanghang Zhang, Guanhong Wu, José M F Moura, João Costeira, and Geoffrey J Gordon. Leveraging multiple source domains for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):652–668, 2020.
- [33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

- [34] Robert Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *California Law Review*, 2019.
- [35] General data protection regulation (gdpr), 2016. Regulation (EU) 2016/679.
- [36] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *ACL*, 2019.
- [37] Proposal for a regulation laying down harmonised rules on artificial intelligence (ai act), 2024. European Parliament provisional agreement.
- [38] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29:2234–2242, 2016.
- [39] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.
- [40] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, 2021.
- [41] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations*, 2017.
- [42] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.
- [43] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *International Conference on Learning Representations*, 2018.
- [44] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 2017.
- [45] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.
- [46] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

- [47] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [48] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 2017.
- [49] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [50] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.
- [51] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [52] Tero Karras, Samuli Laine, and Timo Aila. Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
- [53] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.
- [54] Yunpeng Wang, Meng Pang, Shengbo Chen, and Hong Rao. Consistency-gan: Training gans with consistency model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15743–15751, 2024.
- [55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [56] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- [57] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.
- [58] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photo-realistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

- [59] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Stefano Ermon, and Ben Poole. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. URL <https://arxiv.org/abs/2303.01469>.
- [60] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- [61] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.
- [62] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.
- [63] Ankit Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. *20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.
- [64] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

