



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Redes neuronales generativas antagónicas para mejorar la calidad de sistemas de identificación facial

Informe de Proyecto de Grado presentado por

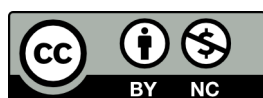
Ana Paula Mazzeo Rodríguez y María Agustina Mazzeo
Rodríguez

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de
la República

Supervisor

Sergio Nesmachnow

Montevideo, 19 de diciembre de 2025



Redes neuronales generativas antagónicas para mejorar la calidad de sistemas de identificación facial por Ana Paula Mazzeo Rodríguez y María Agustina Mazzeo Rodríguez tiene licencia [CC Atribución - No Comercial 4.0](https://creativecommons.org/licenses/by-nc/4.0/).

Agradecimientos

Queremos expresar nuestro agradecimiento a todas las personas e instituciones que hicieron posible este proyecto de grado.

En primer lugar, agradecemos a Sergio Nesmachnow, cuya orientación constante y disposición fueron fundamentales para el desarrollo del proyecto.

Asimismo, valoramos especialmente el apoyo del equipo de ClusterUY. La ejecución de los experimentos presentados en este proyecto se realizó gracias a su infraestructura y al soporte técnico sostenido que su equipo proporcionó en todo momento.

También queremos agradecer a todos los profesores que formaron parte de nuestra formación académica. Nos llevamos experiencias y aprendizajes que marcaron nuestro recorrido en la facultad.

Y, por último, queremos agradecer a nuestra familia, que nos acompañó y acompaña siempre.

A todos quienes colaboraron directa o indirectamente, muchas gracias por su apoyo.

Resumen

El proyecto se centra en la generación de rostros humanos sintéticos mediante un modelo de Red Generativa Antagónica, abordando la necesidad de producir imágenes realistas y controladas en cuanto a sus atributos faciales. El objetivo principal es desarrollar un algoritmo evolutivo para explorar el espacio latente del generador y obtener rostros que combinen simultáneamente similitud con un individuo de referencia, evaluados con dos reconocedores faciales independientes, y correspondencia con una raza específica. La solución implementa un algoritmo evolutivo multiobjetivo que guía el proceso de generación mediante la evaluación iterativa de las imágenes sintéticas en función de los atributos especificados. El algoritmo utiliza en su función de evaluación StyleGAN3 como el generador de imágenes de rostros sintéticos, los modelos de reconocimiento VGG-Face y Dlib para evaluar la similitud facial y el clasificador de raza FairFace para evaluar la correspondencia racial. Los resultados indican que las soluciones producidas presentan un rendimiento consistente, incluso cuando se analizan con diferentes reconocedores, demostrando la capacidad del algoritmo para mantenerse robusto frente a diversos criterios de evaluación. El algoritmo evolutivo desarrollado demostró su capacidad para generar rostros con atributos de raza específicos que engañaron a los reconocedores VGG-Face y Dlib, produciendo resultados efectivos para todos los objetivos evaluados.

Palabras clave: Reconocimiento facial, Redes generativas antagónicas, Algoritmos evolutivos, VGG-Face, Dlib, StyleGAN3, Sesgo

Índice general

1. Introducción	1
2. Marco teórico	3
2.1. Aprendizaje automático	3
2.1.1. Redes neuronales artificiales	3
2.2. Algoritmos evolutivos	4
2.2.1. Funcionamiento general de los algoritmos evolutivos	4
2.2.2. Espacio de búsqueda y espacio de soluciones	5
2.2.3. Algoritmos genéticos	6
2.2.4. Función de fitness	6
2.2.5. Operadores evolutivos	6
2.2.6. Algoritmos evolutivos multiobjetivo	8
2.3. Redes neuronales generativas antagónicas	10
2.3.1. Introducción	10
2.3.2. Espacio latente	11
2.4. Reconocimiento facial	11
2.4.1. Principios de funcionamiento del reconocimiento facial	11
2.4.2. DeepFace como herramienta de reconocimiento facial	12
3. Presentación del problema y trabajos relacionados	13
3.1. Presentación del problema	13
3.2. Trabajos relacionados	14
3.2.1. Motivación	14
3.2.2. Generación de rostros a través del uso de EAs para la exploración del espacio latente de GANs	15
3.2.3. Estado del arte de los sistemas de reconocimiento facial	19
3.2.4. Sesgo en los conjuntos de datos de imágenes faciales	19
3.2.5. Resumen de los artículos relevados	20
4. Arquitectura de la solución	23
4.1. Estructura general de la solución	23
4.2. Algoritmo evolutivo	24
4.2.1. Representación de los individuos	24
4.2.2. Función de evaluación	24

4.2.3.	Operadores evolutivos	25
4.2.4.	Parámetros generales del algoritmo	26
4.3.	Generación de rostros sintéticos	26
4.4.	Reconocimiento facial	27
4.4.1.	Estructura general del sistema de reconocimiento	27
4.4.2.	Biblioteca y modelos utilizados	28
4.5.	Análisis de raza	30
4.5.1.	FairFace	30
4.6.	Modificaciones implementadas para optimizar la eficiencia computacional	31
4.7.	Funcionamiento del pipeline completo	32
5.	Metodología	35
5.1.	Metodología para selección de modelos de detección y reconocimiento	35
5.2.	Metodología para el análisis de los sistemas de reconocimiento	36
5.2.1.	Metodología para la ponderación de distancias entre sistemas de reconocimiento facial	36
5.2.2.	Metodología para el cálculo experimental de los umbrales	36
5.3.	Metodología para la sustitución del reconocedor de raza	37
5.3.1.	Metodología para la selección del nuevo reconocedor de raza	37
5.3.2.	Metodología para evaluación del clasificador de raza modificado	38
5.3.3.	Metodología para el cálculo del umbral de FairFace	39
5.4.	Metodología para el análisis de sesgo en la generación y evaluación de rostros	39
5.5.	Metodología para la evaluación del algoritmo propuesto	39
5.5.1.	Elección de rostros	40
5.5.2.	Métricas utilizadas para la evaluación	42
5.5.3.	Elección del tamaño de población	43
5.5.4.	Elección del número de generaciones	44
5.5.5.	Elección de la configuración paramétrica	44
5.5.6.	Metodología para la validación	45
6.	Experimentación	47
6.1.	Análisis de los modelos de reconocimiento	47
6.1.1.	Ponderación de distancias entre modelos de reconocimiento facial	47
6.1.2.	Cálculo experimental de los umbrales de reconocimiento	48
6.2.	Evaluación de modelos de clasificación de atributos de raza	51
6.2.1.	Evaluación con FairFace	51
6.2.2.	Comparación de DeepFace y FairFace	51
6.2.3.	Cálculo experimental del umbral de FairFace	52
6.2.4.	Evaluación del sesgo en la generación de rostros con StyleGAN3	55
6.3.	Evaluación del algoritmo multiobjetivo	56

<i>ÍNDICE GENERAL</i>	IX
6.3.1. Experimentación preliminar	56
6.3.2. Configuración paramétrica	58
6.3.3. Validación	60
7. Conclusiones y trabajo futuro	69
7.1. Conclusiones	69
7.2. Trabajo futuro	72
Referencias	73
A. Tablas de configuración paramétrica	77
A.1. Experimentación preliminar	77
A.2. Resultados de configuración paramétrica	80
B. Validación	85
B.1. Evolución del hipervolumen	85
B.2. Evolución del frente de Pareto	87
B.3. Soluciones no dominadas	89
B.4. Calidad de resultados	98

Capítulo 1

Introducción

El reconocimiento facial se ha establecido como una tecnología esencial en ámbitos como la seguridad, la autenticación biométrica y en tareas cotidianas, como el desbloqueo de dispositivos móviles (Mohsienuddin y Sabri, 2020). En paralelo, la generación de rostros humanos sintéticos se ha convertido en un área esencial para estudiar y mejorar los sistemas de reconocimiento facial, ya que permite analizar sesgos demográficos, crear conjuntos de datos más equilibrados y explorar escenarios que resultan difíciles de obtener mediante datos reales. Las redes generativas antagónicas (GANs) han demostrado ser especialmente eficaces para este propósito, al producir imágenes faciales de alta calidad y realismo (Karras y cols., 2021).

Sin embargo, controlar de manera precisa los atributos presentes en las imágenes producidas es un gran desafío. La estructura de alta dimensionalidad del espacio latente de las GANs dificulta predecir cómo ciertos desplazamientos o manipulaciones del vector de entrada afectan características específicas del rostro generado. En este contexto, los algoritmos evolutivos ofrecen mecanismos adecuados para explorar espacios de alta dimensionalidad, mantener la diversidad de las soluciones y obtener individuos que se adapten a múltiples objetivos.

Este proyecto propone la implementación de un algoritmo evolutivo multi-objetivo (MOEA) integrado con modelos generativos, de reconocimiento facial y de clasificación racial, para la generación de rostros sintéticos con rasgos raciales preestablecidos. Se continúa con la línea de trabajo comenzada por Machín y cols. (2021) y continuada por Correa y cols. (2023), quienes comprobaron que un enfoque evolutivo es capaz de guiar la búsqueda para la generación de rostros con atributos determinados. A modo de extensión, se propone la utilización de más de un reconocedor facial para la evaluación de similitud facial, que busca disminuir el acoplamiento a un único sistema de reconocimiento. Adicionalmente, se incorpora un clasificador de atributos de raza entrenado con un conjunto de datos demográficamente equilibrado que busca disminuir el sesgo reportado en trabajos anteriores.

La combinación de los componentes de similitud facial y clasificación de raza busca evaluar la utilidad de las imágenes generadas para el análisis del sesgo y la construcción de conjuntos de datos demográficamente más representativos para estudiar el comportamiento de los sistemas de reconocimiento facial ante variaciones sistemáticas en los atributos faciales.

Los resultados obtenidos evidencian que el MOEA propuesto es capaz de generar rostros sintéticos que cumplen con los objetivos de similitud facial y clasificación de raza. Además, el uso de más de un reconocedor facial contribuyó a una evaluación más robusta, reduciendo el acoplamiento a un único sistema de reconocimiento y permitiendo obtener imágenes sintéticas más fiables para el análisis del sesgo y la construcción de conjuntos de datos representativos.

El documento presenta la siguiente estructura. El [Capítulo 2](#) desarrolla el marco teórico del proyecto, incluyendo los fundamentos de la inteligencia computacional, las redes generativas antagónicas, los algoritmos evolutivos y las etapas de un sistema de reconocimiento facial. El [Capítulo 3](#) define con precisión el problema abordado y revisa trabajos relacionados con la generación de rostros sintéticos mediante GANs integradas en un algoritmo evolutivo. El [Capítulo 4](#) describe la arquitectura completa del pipeline, detallando la interacción entre los módulos de generación, reconocimiento facial y evaluación de atributos dentro del algoritmo evolutivo. El [Capítulo 5](#) expone la metodología de experimentación empleada en el proyecto y el procedimiento de los experimentos realizados sobre las configuraciones de parámetros del algoritmo evolutivo. El [Capítulo 6](#) presentan los resultados de la experimentación realizada. Finalmente, el [Capítulo 7](#) presentan las conclusiones y el trabajo futuro.

Capítulo 2

Marco teórico

Este capítulo presenta los conceptos teóricos que sustentan el desarrollo de este proyecto de grado. En primer lugar, se aborda el aprendizaje automático, que proporciona la base teórica sobre la cual se desarrolla el proyecto. A continuación, se introducen los algoritmos evolutivos como estrategia de exploración y optimización. Posteriormente, se describe el funcionamiento de las redes generativas antagónicas. Finalmente, se presentan las principales técnicas de reconocimiento facial, analizando sus componentes, funcionamiento, bibliotecas y modelos de relevancia en el contexto del proyecto.

2.1. Aprendizaje automático

El aprendizaje automático es una subdisciplina de la inteligencia computacional dedicada al estudio y desarrollo de algoritmos capaces de mejorar su desempeño en tareas específicas a partir de la experiencia. A diferencia de las técnicas basadas en programación tradicional, los algoritmos de aprendizaje automático se diseñan para buscar comportamientos similares a partir de patrones identificados en conjuntos de datos o en experiencia previa, sin necesidad de definir reglas explícitas para cada caso posible (T. Mitchell, 1997). En la actualidad, el aprendizaje automático ocupa un papel central en la resolución de problemas complejos y se aplica en campos como el procesamiento del lenguaje natural, la bioinformática y la visión por computadora (Goodfellow y cols., 2016).

2.1.1. Redes neuronales artificiales

Las redes neuronales artificiales (ANNs) constituyen una categoría de modelos dentro del aprendizaje automático cuyo diseño está inspirado en el funcionamiento del sistema nervioso humano. Las ANNs se estructuran mediante un conjunto de nodos interconectados organizados en capas diferenciadas: una capa inicial de entrada, una o múltiples capas ocultas intermedias y una capa final de salida.

Cada nodo procesa un conjunto de entrada, donde cada componente es integrado linealmente mediante parámetros ponderados o pesos sinápticos. Posteriormente, el resultado de la combinación lineal es transformado por una función de activación no lineal, generando así una salida del nodo. El proceso de aprendizaje en estas redes consiste en ajustar iterativamente los pesos con el fin de minimizar una función objetivo o de pérdida, encargada de cuantificar el error cometido por la red neuronal.

Una arquitectura particularmente relevante dentro del campo de las redes neuronales artificiales es la constituida por las redes neuronales convolucionales (CNNs), especializadas para trabajar con datos que poseen estructura espacial, como es el caso particular de las imágenes (LeCun y cols., 2015). Las CNNs se componen principalmente de tres tipos de capas. La capa convolucional utiliza filtros que se aplican sobre pequeñas regiones de la entrada. Los filtros se van desplazando para procesar progresivamente todas las regiones de la entrada y generar múltiples mapas de características, cada uno de los cuales captura diferentes características de la entrada. Luego, la CNN aplica la función de activación ReLU a los mapas de características. La capa de agrupación reduce la dimensionalidad conservando las características esenciales de la entrada. En la capa totalmente conectada, cada nodo de la capa de salida está conectado al nodo de la capa anterior. La capa generalmente utiliza una función de activación softmax que transforma un vector en una distribución de probabilidad y genera una salida normalizada. Las CNNs se han convertido en una herramienta fundamental para tareas de clasificación de imágenes, detección de objetos y reconocimiento facial.

2.2. Algoritmos evolutivos

Esta sección ofrece una visión detallada de los algoritmos evolutivos. Se explica su funcionamiento y se describen sus componentes principales, como la función de fitness y los operadores evolutivos. Asimismo, se definen los algoritmos genéticos, se describen conceptos clave y se explica la distinción entre el espacio de búsqueda y el espacio de soluciones. Finalmente, se aborda la extensión a la optimización multiobjetivo, ilustrada con el algoritmo NSGA-II.

2.2.1. Funcionamiento general de los algoritmos evolutivos

Los algoritmos evolutivos (EAs) pertenecen a una familia de métodos computacionales inspirados en los principios biológicos de la evolución natural. Los EAs permiten resolver problemas complejos de optimización mediante mecanismos específicos, incluidos la selección natural, la mutación y el cruzamiento.

Los EAs inician con la generación de una población inicial de individuos. Cada individuo es la representación de una posible solución al problema de optimización que se aborda. Los individuos se someten a una evaluación mediante una función de fitness que cuantifica su calidad (Eiben y Smith, 2015). El al-

goritmo selecciona los individuos con mayor probabilidad de ser elegidos como padres y aplica operadores de cruzamiento y mutación para generar los nuevos descendientes que formarán la siguiente generación. El ciclo de evaluación, selección, cruzamiento y mutación se repite durante un número determinado de generaciones o hasta que se cumpla un criterio de parada, como converger a un umbral de fitness predeterminado.

La efectividad de los EAs radica en su capacidad de explorar simultáneamente múltiples soluciones potenciales, promoviendo así la adaptabilidad y diversidad en los resultados obtenidos. Dado que los EAs trabajan con una población de soluciones, son una herramienta particularmente adecuada para abordar problemas de optimización complejos, donde los métodos tradicionales pueden presentar limitaciones, especialmente cuando los problemas involucran múltiples objetivos (Deb, 2001). El Algoritmo 1 presenta un pseudocódigo de un EA genérico.

Algoritmo 1 Algoritmo evolutivo

```
1: Iniciar población con soluciones candidatas aleatorias;
2: Evaluar cada candidato;
3: repeat
4:   Seleccionar padres;
5:   Recombinar parejas de padres;
6:   Mutar la descendencia resultante;
7:   Evaluar los nuevos candidatos;
8:   Seleccionar individuos para la siguiente generación;
9: until se cumpla la condición de terminación
10: Fin
```

2.2.2. Espacio de búsqueda y espacio de soluciones

Los individuos de un algoritmo evolutivo son representaciones de soluciones potenciales al problema que resuelve. Las soluciones potenciales existen dentro de un espacio de soluciones o espacio de fenotipos. El conjunto de representaciones de soluciones, que son manipuladas por el EA, se denomina espacio de búsqueda o espacio de genotipos (Eiben y Smith, 2015).

La codificación de soluciones constituye un aspecto fundamental en la implementación de los EAs, siendo el componente crítico que establece la conexión entre el espacio de soluciones y el espacio de búsqueda. La codificación define específicamente cómo quedan representadas las soluciones potenciales dentro del espacio de búsqueda, estableciendo el mapeo conocido como genotipo-fenotipo (Bartz-Beielstein y cols., 2014). Generalmente, las soluciones se codifican mediante vectores numéricos o cadenas binarias, aunque también existen variantes que utilizan representaciones más complejas como grafos, árboles o matrices. La elección del tipo de codificación suele depender de la naturaleza específica del problema a resolver. Una buena elección de la representación afecta significativamente la eficiencia y efectividad del EA.

2.2.3. Algoritmos genéticos

Dentro de la familia de algoritmos evolutivos, los algoritmos genéticos (AGs) son el tipo de algoritmo evolutivo más conocido. [M. Mitchell \(1999\)](#) comentaba que la primera idea de AGs fue propuesta y desarrollada por John Holland en la década de 1970, con el objetivo de trasladar los mecanismos evolutivos a sistemas informáticos.

Los AGs utilizan una población de soluciones codificadas que evolucionan mediante selección natural junto con operadores de cruzamiento y mutación. En el contexto específico de los AGs, las soluciones codificadas reciben el nombre de cromosomas, el genotipo que manipula directamente el algoritmo. Cada cromosoma está típicamente representado como una cadena de bits compuesta de genes, donde cada gen toma valores de 0 o 1.

2.2.4. Función de fitness

La función de fitness es uno de los componentes más importantes de los algoritmos evolutivos y establece el vínculo fundamental entre el problema a resolver y el EA. La función de fitness define una medida cuantitativa de calidad para cada solución candidata a partir de su fenotipo. Esta función cumple el rol de orientar el proceso selectivo. A partir de los valores de fitness se determinan los individuos que tienen mayor probabilidad de ser elegidos como progenitores ([Eiben y Smith, 2015](#)).

La eficiencia computacional asociada a la evaluación de la función de fitness es importante para la viabilidad práctica del algoritmo. Dado que los algoritmos evolutivos trabajan con poblaciones, implica que en cada generación deben evaluarse múltiples individuos. Por ejemplo, para una población de 100 individuos con una función de fitness que requiere una hora para evaluar a cada uno, la tarea de completar 100 generaciones implica, a lo sumo, 10,000 evaluaciones en total, lo que se traduce en aproximadamente 1,14 años de tiempo de procesamiento ([Whitley, 1994](#)).

2.2.5. Operadores evolutivos

Esta sección presenta brevemente los operadores evolutivos probabilísticos que se utilizan en los EAs.

Selección

La selección es el proceso que determina qué individuos de la población actual serán elegidos para crear la siguiente generación. Su principal objetivo es privilegiar a aquellos individuos con mayor aptitud, otorgándoles una probabilidad de reproducción superior y dirigiendo así el proceso evolutivo hacia soluciones de mayor calidad ([Eiben y Smith, 2015](#)).

En este contexto, la presión de selección se refiere a la intensidad con la que el mecanismo de selección favorece a los individuos más aptos frente a los menos aptos en cada generación. Una presión de selección alta acelera la propagación de

rasgos favorables en la población, pero tiene como desventaja que puede reducir la diversidad genética, conduciendo a una convergencia prematura en óptimos locales. Por el contrario, una presión de selección baja facilita la exploración de nuevas regiones del espacio de búsqueda, aunque el progreso hacia soluciones de mayor calidad puede ser más lento.

Existen diversos métodos de selección, como la selección por torneo, donde se busca seleccionar λ individuos. Para ello, se elige aleatoriamente un pequeño subconjunto de k individuos para que compitan entre ellos y seleccionar al individuo con mejor fitness. Este proceso se repite hasta obtener los λ individuos. Otro método es la selección proporcional al fitness, donde el valor de la función de fitness de cada individuo determina de manera directa la probabilidad de ser seleccionado.

Cruzamiento

El cruzamiento combina la información genética de dos o más progenitores para crear uno o más descendientes potencialmente mejores y permite la explotación de buenas regiones del espacio de búsqueda. El principio subyacente es que al mezclar las características de individuos con mayor aptitud es posible generar descendientes que heredan una combinación superior de sus rasgos (Eiben y Smith, 2015). El cruzamiento opera sobre la representación genética (genotipo) y se aplica a un conjunto de la población de padres de forma aleatoria con una probabilidad de cruzamiento p_c .

Un ejemplo común es el cruzamiento de un punto, que opera eligiendo un punto de corte aleatorio y dividiendo a los dos padres en ese punto. Luego, los segmentos resultantes se intercambian entre sí para formar dos nuevos descendientes. Un ejemplo del funcionamiento del cruzamiento de un punto se ilustra en la Figura 2.1.

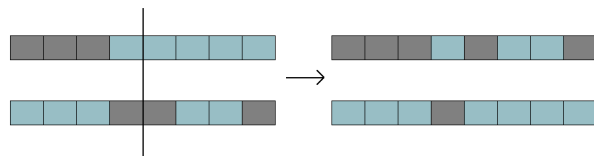


Figura 2.1: Ilustración del funcionamiento del cruzamiento de un punto.

Mutación

La mutación es un operador que introduce cambios pequeños y aleatorios en el genotipo de un individuo. Su papel es fundamental para la exploración del espacio de búsqueda, mantener la diversidad genética y prevenir el estancamiento del algoritmo en óptimos locales (Eiben y Smith, 2015).

Para una representación binaria, el operador de mutación más simple es el bit-flip, que consiste en seleccionar un bit aleatorio i e invertir su valor. La Figura 2.2 ejemplifica el funcionamiento de la mutación bit-flip.

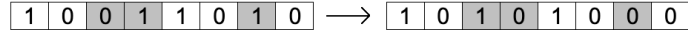


Figura 2.2: Ilustración del funcionamiento de mutación bit-flip.

2.2.6. Algoritmos evolutivos multiobjetivo

La optimización multiobjetivo se enfoca en resolver problemas que requieren la optimización simultánea de varios objetivos. A diferencia de los problemas monoobjetivo, donde generalmente existe una única solución óptima, en los problemas multiobjetivo las funciones a optimizar suelen estar en conflicto entre sí donde la mejora en un objetivo puede implicar el deterioro en otro. En este marco, se introduce el concepto de dominancia: una solución domina a otra si es igual o superior en todos los objetivos y estrictamente mejor en al menos uno de ellos. En consecuencia, una solución es no dominada si no existe ninguna otra solución en el espacio de búsqueda que la domine. El resultado de la optimización multiobjetivo es un conjunto de soluciones no dominadas. El conjunto de todas las soluciones no dominadas conforma el frente de Pareto.

El principio de la optimización multiobjetivo se caracteriza por dos objetivos complementarios. Por un lado, se busca la convergencia de las soluciones hacia el frente de Pareto, y por otro lado, se busca mantener una diversidad suficiente entre las soluciones obtenidas para que el conjunto resultante cubra de manera homogénea todo el rango de posibles compromisos entre los objetivos.

Los EAs se presentan como un paradigma de búsqueda especialmente adecuado para resolver problemas de optimización multiobjetivo debido a su naturaleza basada en poblaciones. La capacidad de un EA para procesar y evolucionar un conjunto diverso de soluciones candidatas en cada generación permite encontrar múltiples soluciones no dominadas en una única ejecución (Bartz-Beielstein y cols., 2014; Deb, 2001).

NSGA-II

El algoritmo Non-dominated Sorting Genetic Algorithm (NSGA-II) constituye uno de los algoritmos evolutivos multiobjetivo más efectivos en la literatura. El algoritmo NSGA-II se destaca por su robusta capacidad para mantener la diversidad entre las soluciones encontradas, abordando los dos propósitos fundamentales de la optimización multiobjetivo: la convergencia hacia el frente de Pareto y la distribución uniforme de las soluciones a lo largo del frente. El éxito del algoritmo reside en la integración de un principio elitista que garantiza la preservación de las mejores soluciones, con un mecanismo explícito para la preservación de la diversidad y un procedimiento de selección que prioriza las soluciones no dominadas (Deb, 2001).

El algoritmo NSGA-II inicia su ejecución generando de manera aleatoria una población inicial de N individuos. Posteriormente, se evalúan las funciones objetivo que el algoritmo busca minimizar o maximizar simultáneamente.

Con los valores resultantes de las funciones objetivo se aplica el procedimiento de ordenamiento no dominado sobre la población inicial. El procedimiento clasifica las soluciones en múltiples frentes según sus relaciones de dominancia y asigna a cada solución un rango que refleja su nivel de dominancia. Las soluciones que no son dominadas por ninguna otra conforman el primer frente. Una vez identificado el primer frente, se excluye temporalmente de la población y se repite el procedimiento para identificar el segundo frente. El proceso iterativo continúa hasta que todos los individuos han sido clasificados en algún frente. Luego inicia el ciclo iterativo donde en cada iteración t se realiza la selección de padres P_t , generalmente a través de la selección por torneo. Los padres seleccionados se someten a los operadores genéticos, como el cruzamiento y la mutación, para producir una población de descendientes Q_t de tamaño N .

Para formar la población de la siguiente generación, se combinan la población de padres y la población de descendientes que forman una población unificada de tamaño doble R_t . Sobre la población combinada R_t se aplica nuevamente el procedimiento de ordenamiento no dominado. La nueva población se construye seleccionando individuos comenzando por el frente de mejor rango. Los frentes completos se aceptan sucesivamente hasta que la inclusión del siguiente frente completo excede el tamaño de población deseado. Cuando no todos los individuos del último frente considerado pueden ser incluidos, se aplica la distancia de crowding como criterio de desempate.

La distancia de crowding constituye una medida que aproxima el volumen del cuboide que rodea a un individuo, usando como límites a sus vecinos más cercanos, como muestra en la [Figura 2.3](#). Para los individuos intermedios, la medida se obtiene como la suma de las distancias normalizadas entre los valores de los objetivos de los individuos adyacentes. La [Ecuación 2.1](#) muestra cómo se calcula la distancia de crowding para un individuo i , siendo M el número de objetivos y $f_m^{\max}$ y $f_m^{\min}$ los valores máximo y mínimo del objetivo m dentro del frente considerado. Una mayor distancia de crowding es preferible porque indica que el individuo se encuentra en una región menos poblada del frente considerado. De esta forma se contribuye a mantener de la diversidad de soluciones.

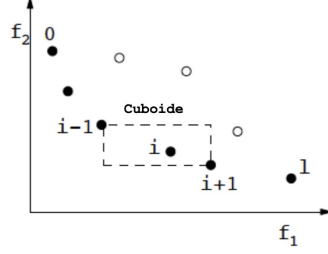


Figura 2.3: Cuboide creado para el cálculo de la distancia de crowding.

$$CD_i = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{\text{máx}} - f_m^{\text{mín}}} \quad (2.1)$$

2.3. Redes neuronales generativas antagónicas

Esta sección presenta los conceptos principales de las redes generativas antagónicas y define el espacio latente.

2.3.1. Introducción

Las redes neuronales generativas antagónicas (GANs) pertenecen a los modelos generativos basados en aprendizaje profundo. Las GANs se caracterizan por su capacidad de generar datos sintéticos realistas mediante un proceso de entrenamiento competitivo entre un modelo generativo (o generador) y un modelo discriminatorio (o discriminador).

Las GANs fueron introducidas por [Goodfellow y cols. \(2014\)](#) como un marco de aprendizaje no supervisado en el que dos redes neuronales profundas se enfrentan en un juego de suma cero. El generador G toma como entrada un vector de ruido z proveniente de una distribución latente $p_z(z)$ y produce datos sintéticos $G(z)$. El discriminador D , recibe como entrada datos reales $x \sim p_{data}(x)$ y datos generados $G(z)$ y su objetivo es distinguir entre ambos. El entrenamiento se plantea como un problema de optimización min-max, donde el generador intenta maximizar el error cometido por el discriminador, y el discriminador intenta minimizarlo.

Durante el entrenamiento, el generador y el discriminador se optimizan de manera alternada. En cada iteración, primero se ajustan los parámetros del discriminador con el objetivo de mejorar su capacidad para distinguir entre muestras reales y sintéticas. Luego se actualizan los parámetros del generador para reducir la eficacia del discriminador en esa tarea, es decir, G se entrena para producir muestras que el discriminador clasifique como reales.

El objetivo final del entrenamiento es alcanzar un equilibrio en el que el generador produzca muestras sintéticas tan verosímiles que el discriminador no pueda distinguirlas de las reales con un nivel de certeza superior al azar. Esto implica que, en el óptimo teórico, la distribución generada es igual a la distribución de los datos reales.

2.3.2. Espacio latente

En el contexto de las GANs, el espacio latente es una representación matemática abstracta y multidimensional en la que se codifican los vectores que sirven como entrada al generador de una GAN. Cada vector es una representación de las características visuales de una imagen. A partir de un vector del espacio latente el generador produce una imagen sintética única, de modo que variaciones en las coordenadas del vector producen cambios visibles en la imagen sintética resultante.

Sin embargo, el espacio latente no posee una interpretación directa. El entrenamiento del generador no especifica restricciones sobre como debe interpretarse el vector de entrada. En consecuencia, es posible que el generador emplee esta información de un modo altamente entrelazado, de forma tal que las coordenadas del vector no correspondan directamente con las características semánticas de los datos. Comprender cómo se codifican las características faciales dentro del espacio latente posibilita la generación de rostros con características específicas deseadas.

2.4. Reconocimiento facial

Esta sección describe el funcionamiento general de un sistema de reconocimiento facial, detalla sus etapas fundamentales y presenta algunas de las herramientas más utilizadas para su implementación.

2.4.1. Principios de funcionamiento del reconocimiento facial

El funcionamiento de un sistema de reconocimiento facial moderno se estructura típicamente en cuatro etapas principales: detección, alineación, representación y verificación (Taigman y cols., 2014). Las fases componen un flujo de procesamiento secuencial que permite identificar personas en imágenes.

La detección facial es la etapa inicial y se encarga de localizar una o más regiones faciales dentro de una imagen. En esta etapa, se descartan áreas no faciales, lo que ayuda a evitar sesgos provocados por el ruido en los resultados. En general, los detectores proporcionan cajas o elipses delimitadoras del rostro y además identifican puntos característicos de un rostro como la posición de los ojos, la nariz o la boca. La alineación facial es la segunda etapa, crucial para mejorar los resultados del sistema frente a variaciones de pose, iluminación y expresión. La alineación busca normalizar la apariencia del rostro detectado. En

su forma más simple, rota el rostro para dejar los ojos horizontales. La tercera etapa es la representación que transforma el rostro alineado en un vector de características o embedding de dimensión fija que captura la identidad. En la práctica, esta etapa se implementa con CNNs entrenadas para producir descriptores compactos y robustos. De esta forma, los vectores obtenidos al procesar dos fotografías del mismo individuo quedan más cercanos en el espacio vectorial que los generados a partir de rostros de personas distintas. En la etapa de verificación, los vectores obtenidos en la etapa de representación se comparan para determinar si pertenecen a la misma identidad. La similitud entre ambos vectores se cuantifica mediante métricas como la distancia euclidiana o el coseno. Finalmente, el valor de distancia obtenido se evalúa contra un umbral previamente determinado para decidir si se trata de la misma persona o personas distintas (Serengil y Ozpinar, 2024).

El reconocimiento en fotografías se aplica extensamente en contextos como control de acceso, vigilancia, seguridad y verificación de identidad. En este marco, las etapas de detección y alineación de un sistema de reconocimiento facial resultan cruciales para aportar robustez frente a las variaciones estáticas de las imágenes como pose, expresión facial, iluminación y calidad de captura.

2.4.2. DeepFace como herramienta de reconocimiento facial

Una implementación práctica sistemas de reconocimiento facial ampliamente utilizada es DeepFace, desarrollada por Serengil y Ozpinar (2020) como una biblioteca de código abierto escrita en Python. DeepFace integra múltiples modelos de detección y reconocimiento facial de estado del arte y proporciona una interfaz sencilla y funcional para tareas de verificación de identidad, análisis de atributos faciales y reconocimiento facial en imágenes y videos.

La biblioteca DeepFace permite acceder a diversos modelos de reconocimiento facial preentrenados: VGG-Face, FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib, SFace, GhostFaceNet y Buffalo L. También proporciona múltiples modelos de detección para las primeras etapas de un sistema de reconocimiento: OpenCV, SSD, Dlib, MTCNN, Faster MTCNN, RetinaFace, MediaPipe, YOLO, YuNet y CenterFace. Estos componentes pueden combinarse para obtener diferentes sistemas de reconocimiento facial.

Capítulo 3

Presentación del problema y trabajos relacionados

Este capítulo expone el problema que aborda este proyecto de grado, describe su contexto, motivación y principales enfoques de resolución. Asimismo, presenta una revisión de la literatura relevante en la generación de rostros utilizando GANs y algoritmos evolutivos, el estado del arte en sistemas de reconocimiento facial y estudios sobre sesgo racial en conjuntos de datos de imágenes faciales.

3.1. Presentación del problema

Las redes generativas antagónicas operan sobre un espacio vectorial, o espacio latente, de alta dimensionalidad. En el espacio latente, cada vector representa una codificación única de atributos que el generador transforma en una imagen única. Una propiedad del espacio latente es que vectores cercanos producen instancias con características similares. Saber cuáles son las regiones del espacio que generan las características que resuelven un determinado problema es un gran desafío. Se necesita garantizar la diversidad de soluciones, evitar explorar regiones de baja calidad o irrelevantes y, al mismo tiempo, se requiere guiar la búsqueda hacia opciones con características deseadas. Dada la alta dimensionalidad del espacio latente, enfoques tradicionales de optimización como la búsqueda aleatoria o el descenso por gradiente resultan ineficientes y poco escalables.

En este contexto, los algoritmos evolutivos surgen como una estrategia para la exploración del espacio latente, ya que permiten mantener la diversidad con una población de soluciones. Al trabajar con múltiples individuos en paralelo, los algoritmos evolutivos pueden recorrer diferentes regiones del espacio latente de forma simultánea, minimizando el riesgo de quedar estancados en mínimos locales y promoviendo una cobertura más amplia del espacio.

El enfoque de este proyecto consiste en implementar un MOEA que orienta la generación de imágenes de rostros a través de la exploración del espacio latente. El algoritmo evolutivo propuesto busca optimizar dos objetivos independientes. Por un lado, se busca maximizar la similitud entre la imagen generada y una imagen objetivo. Por otro lado, se procura que la imagen generada cumpla con los atributos de una raza predefinida.

Para este proyecto, el primer objetivo tiene la particularidad de que la similitud no se mide con un único sistema de reconocimiento facial, sino con dos reconocedores completamente distintos. En este marco, se busca aportar mayor robustez y reducir el riesgo de obtener soluciones demasiado acopladas a una única implementación de un sistema de reconocimiento facial. El segundo objetivo se evalúa mediante un clasificador de raza entrenado con un conjunto de datos demográficamente equilibrado, lo que permite mitigar el sesgo asociado a la clasificación de la raza.

Los dos objetivos que guían la generación no siempre evolucionan en la misma dirección. El uso de un MOEA permite evaluar cada objetivo de forma independiente y construir un conjunto diverso de soluciones no dominadas que refleja distintos compromisos entre similitud facial y clasificación racial.

La generación sintética de rostros humanos constituye una herramienta clave para evaluar la robustez de los sistemas de reconocimiento facial. Las imágenes sintéticas pueden ser utilizadas para construir grandes conjuntos de datos equilibrados sin recurrir a información de personas reales. De esta forma, se preserva la privacidad de las personas.

Del mismo modo, las imágenes generadas sintéticamente permiten evaluar los reconocedores faciales bajo condiciones experimentales controladas, lo que favorece el desarrollo de sistemas más precisos, equitativos y capaces de mantener su desempeño ante variaciones en las imágenes de entrada.

3.2. Trabajos relacionados

Esta sección presenta la investigación de los trabajos relacionados relevantes para el desarrollo de este proyecto.

3.2.1. Motivación

El propósito de la revisión de trabajos relacionados es investigar el estado del arte en la exploración del espacio latente de redes generativas antagónicas mediante estrategias evolutivas, con énfasis en la generación de rostros artificiales de alta calidad. En particular, se analizan propuestas monoobjetivo y multiobjetivo orientadas a aproximar imágenes objetivo y controlar atributos demográficos. Se busca fundamentar el desarrollo de un MOEA capaz de generar rostros sintéticos que combinen similitud facial con una imagen objetivo y un atributo de raza predefinido, integrando información proveniente de múltiples reconocedores faciales y un clasificador racial entrenado con conjuntos de datos más equilibrados.

3.2.2. Generación de rostros a través del uso de EAs para la exploración del espacio latente de GANs

Esta sección examina trabajos relacionados que integran redes generativas antagónicas con algoritmos evolutivos para la generación de rostros con atributos faciales predeterminados.

Algoritmos evolutivos monoobjetivo

[Machín y cols. \(2021\)](#) desarrollaron un algoritmo evolutivo monoobjetivo como método de búsqueda en el espacio latente de una GAN para generar retratos humanos sintéticos similares a una imagen objetivo. El algoritmo utilizó StyleGAN2 como generador preentrenado de rostros y FaceNet como reconocedor para la evaluación de similitud a partir de las representaciones vectoriales (embeddings) de los rostros. La solución se codificó como un vector de 512 valores reales que representó un punto en el espacio latente de StyleGAN2. La función de fitness se definió como el opuesto de la distancia euclidiana entre los embeddings de FaceNet de la imagen objetivo y la imagen generada. La población inicial se generó de forma aleatoria a partir de una distribución normal estándar. La selección se realizó mediante torneos, el cruzamiento aplicó el operador BLX- α y la mutación se realizó utilizando una distribución gaussiana con media $\mu = 0$ y desviación estándar $\sigma = 1$.

El algoritmo se implementó en Python 3.7 sobre la biblioteca DEAP (versión 1.3.1) e incluyó optimizaciones relevantes para reducir el tiempo de ejecución, como el cálculo de la función de fitness en lotes, la reducción de la resolución de las imágenes (de 1024×1024 a 160×160 píxeles) para la evaluación y la sustitución del modelo de detección por uno más eficiente en tiempo.

Los experimentos mostraron que el algoritmo generó rostros capaces de engañar a FaceNet en menos de quince minutos por instancia y logró reducir la distancia euclidiana hasta en un 27,9 % en comparación con la distancia obtenida de otra imagen real del mismo individuo. Los resultados evidenciaron que el algoritmo presentó mejor desempeño al utilizar imágenes sintéticas como objetivo. En cuanto a la diversidad, el algoritmo produjo conjuntos de retratos suficientemente diversos entre sí pero consistentemente similares al objetivo, lo que evidenció la capacidad de los algoritmos evolutivos para explorar de forma eficaz el espacio latente de GANs y obtener imágenes de alta calidad guiadas por un modelo de reconocimiento facial preentrenado.

En una misma línea de investigación [Machín y cols. \(2022\)](#) presentaron un enfoque evolutivo para generar retratos humanos sintéticos similares a dos rostros objetivos simultáneamente mediante la exploración del espacio latente de una GAN. De la misma forma que en [Machín y cols. \(2021\)](#), el trabajo se apoyó en un modelo StyleGAN2 preentrenado para producir retratos y en FaceNet como modelo de reconocimiento facial para evaluar la similitud.

El algoritmo evolutivo codificó cada solución como un vector de 512 valores reales en el espacio latente de StyleGAN2. Para los operadores, se mantuvo una inicialización aleatoria basada en una distribución normal $N(0, 1)$, selección por

torneo y se empleó el operador de cruzamiento BLX- α junto con la mutación gaussiana. La principal diferencia respecto al algoritmo monoobjetivo anterior fue la modificación de la función de fitness. En este caso, se aplicó un enfoque de agregación lineal de las distancias coseno a cada imagen objetivo (I_i, I_j). La función de fitness se definió como el valor opuesto de la función ponderada $f(\mathbf{x}) = -w_1 \cdot d(\mathbf{x}, I_i) - w_2 \cdot d(\mathbf{x}, I_j)$, donde se utilizó una ponderación ($w_1 = w_2 = 0,5$) para priorizar la captura de características de ambos objetivos.

En la evaluación, se confirmó que la técnica de procesamiento en lotes y la reducción de imágenes para el cálculo de la función de fitness disminuyeron significativamente tiempo de ejecución del algoritmo. La validación del método fue realizada en tres instancias (I_1, I_2, I_3) elegidas con diferentes niveles de dificultad. Los resultados confirmaron que el algoritmo mantuvo la capacidad de generar soluciones de alta calidad. No obstante, se identificó que la tasa de éxito disminuyó para la instancia más desafiante (I_3), donde solo el 38,4% de las soluciones finales lograron engañar a FaceNet para ambos objetivos. En términos de diversidad, el EA generó imágenes variadas, pero la baja dispersión de los resultados sugirió que el EA no fue capaz de generar suficiente diversidad, una limitación que los autores identificaron para futuras investigaciones. En conclusión, los resultados demostraron que el EA fue capaz de generar imágenes de alta calidad que combinaron las características principales de las dos imágenes objetivo.

Los trabajos de Machín y cols. (2021) y Machín y cols. (2022) proporcionan una base conceptual y metodológica sobre el uso de algoritmos evolutivos para explorar espacios latentes de GANs y generar imágenes sintéticas guiadas por sistemas de reconocimiento facial. Su utilidad para este proyecto es evidenciar que la combinación de modelos generativos y algoritmos evolutivos constituye una estrategia eficaz para generar imágenes que cumplan ciertas características deseadas.

Algoritmos evolutivos multiobjetivo

Correa y cols. (2023) se basaron en los trabajos de Machín y cols. (2021) y Machín y cols. (2022) para implementar un MOEA para la generación de imágenes sintéticas de rostros. El objetivo principal de la investigación de Correa y cols. (2023) fue crear ejemplos de imágenes con determinados atributos definidos que fueran capaces de engañar a sistemas de reconocimiento facial. El problema se formuló como la minimización simultánea de dos funciones: la distancia entre la imagen generada y una imagen objetivo y la diferencia entre el atributo racial del rostro generado y el de la imagen objetivo a partir de un vector de probabilidades. A diferencia de los estudios anteriores, el algoritmo utilizó StyleGAN3, una versión mejorada de StyleGAN2, para la generación de rostros sintéticos. Para el reconocimiento el algoritmo empleó la biblioteca DeepFace con el modelo de reconocimiento VGG-Face. Los individuos se representaron con un vector de 512 valores reales que representa un punto en el espacio latente de StyleGAN3. La inicialización de individuos se realizó de manera aleatoria a partir de una distribución normal estándar, aunque para instancias con razas objetivo subre-

presentadas por el generador se implementó un operador de inicialización de individuos inteligente para asegurar que una proporción de la población inicial fuera de la raza deseada. El algoritmo evolutivo multiobjetivo utilizó selección por dominancia de Pareto para la selección de la nueva generación, selección por torneos para la selección de padres y distancia de crowding, cruzamiento BLX- α y mutación gaussiana.

La evaluación consistió en seleccionar seis imágenes provenientes de distintas fuentes: dos imágenes del conjunto de datos LFW, dos imágenes reales externas al conjunto de datos y dos rostros sintéticos generados con StyleGAN2 y StyleGAN3. A cada instancia se le asignó una raza objetivo para guiar la búsqueda hacia soluciones que presentaran el atributo racial asignado. Para evaluar la convergencia del algoritmo se utilizó la evolución de hipervolumen relativo. Cada instancia se sometió a treinta ejecuciones independientes del algoritmo, manteniéndose constante el número de generaciones (100) y el tamaño de la población (100 individuos). Además, se midió la capacidad de las soluciones para engañar al sistema de reconocimiento con VGG-Face y alcanzar la raza objetivo según el clasificador interno de DeepFace. Los resultados mostraron hipervolumenes elevados y buena dispersión en todas las instancias, indicando una aproximación adecuada al frente de Pareto y una diversidad de soluciones no dominadas. Además, en la mayoría de las instancias de validación, más del 90 % de las soluciones no dominadas engañaron al sistema de reconocimiento, aunque el porcentaje de soluciones que lograron la raza objetivo varió considerablemente, por ejemplo, 83,6 % en una instancia y menos del 10 % en otra. Los autores concluyeron que el algoritmo propuesto logró generar imágenes realistas con características específicas, explorar eficazmente el espacio latente de StyleGAN3 y producir ejemplos que desafiaron al sistema de reconocimiento utilizado.

Resulta relevante destacar del artículo de [Correa y cols. \(2023\)](#) que se demuestra la factibilidad de manipular atributos faciales mediante un enfoque evolutivo multiobjetivo y que es capaz de minimizar simultáneamente la similitud con una imagen objetivo y clasificar las soluciones en una raza predeterminada. Además, el uso de StyleGAN3 como generador de rostros sintéticos y la biblioteca de DeepFace proporcionan la estructura fundamental del pipeline construido en este proyecto.

El trabajo de [Nesmachnow y Moreno \(2025\)](#) presentó un enfoque evolutivo multiobjetivo para explorar el espacio latente de StyleGAN3 con el fin de generar rostros humanos que cumplieran simultáneamente tres criterios: similitud con una imagen objetivo, coincidencia de género y coincidencia de raza. La motivación se sustentó en la necesidad de generar muestras capaces de exponer vulnerabilidades de los sistemas de reconocimiento facial y de analizar posibles sesgos de género y raza presentes en los modelos generativos.

En cuanto a la infraestructura, el estudio reutilizó el pipeline desarrollado en trabajos anteriores ([Correa y cols., 2023](#)), con la diferencia de que el modelo utilizado para el reconocimiento fue Facebook DeepFace. Además, la biblioteca DEAP se modificó para permitir el procesamiento por lotes de las imágenes generadas y se aplicaron técnicas de reducción y detección facial lo que redujo significativamente los tiempos de ejecución en la evaluación de aptitud.

El trabajo presenta una comparación de tres variantes de estrategias evolutivas multiobjetivo (MOES): S-MOES, una estrategia estándar que solo utilizó la mutación como operador de variación, H-MOES que añadió un operador de recombinación y E-MOES que extendió la versión híbrida mediante la inclusión de un grid adaptativo y un archivo externo destinado a preservar la diversidad del frente de Pareto. En los tres casos, cada individuo se codificó como un vector de 512 valores reales inicializado mediante muestreo aleatorio de una distribución normal, y se aplicó mutación gaussiana como operador principal. La recombinación se empleó únicamente en las variantes híbridas: cruzamiento en dos puntos en H-MOES y BLX- α en E-MOES. Para la evaluación, los autores emplearon seis imágenes de referencia que representaban diversidad en raza, género y edad. La calidad de los frentes obtenidos se midió mediante dos métricas, hipervolumen relativo y EAS-50 %. Además, se incluyó un método base de optimización, ε -constraint, para contrastar los algoritmos evolutivos con un enfoque tradicional de resolución de problemas multiobjetivo.

Los valores de hipervolumen relativo mostraron que E-MOES obtuvo los mejores frentes de Pareto en cuatro de las seis instancias, mientras que en las restantes mantuvo resultados comparables a H-MOES. En relación con la capacidad de engañar al sistema de reconocimiento, los porcentajes de imágenes que lograron superar simultáneamente los umbrales de similitud, raza y género fueron consistentemente más altos en E-MOES, con tasas que en algunos casos superaron el 90 %. Las imágenes generadas presentaron una diversidad visual considerable y, según los ejemplos reportados, conservaron rasgos coherentes con los atributos objetivo, incluso en instancias difíciles como rostros andróginos o razas subrepresentadas.

El estudio concluyó que las estrategias evolutivas multiobjetivo constituyen un enfoque eficaz para la exploración del espacio latente de StyleGAN3 cuando se requieren simultáneamente múltiples atributos. Entre las variantes propuestas, E-MOES demostró el mejor equilibrio entre convergencia y diversidad y alcanzó el mejor desempeño para generar imágenes que engañaran a Facebook DeepFace respecto de los atributos de similitud, raza y género. Los autores también señalaron diversas limitaciones inherentes al proceso. La generación y evaluación de imágenes sintéticas resultó computacionalmente costosa, incluso con técnicas de reducción de resolución y procesamiento por lotes. Además, la habilidad del pipeline para generar imágenes con los atributos deseados depende de los sesgos existentes en StyleGAN3 y en DeepFace, lo que afectó la precisión para ciertas combinaciones de atributos.

El artículo de [Nesmachnow y Moreno \(2025\)](#) es importante para este proyecto porque demuestra que las estrategias evolutivas multiobjetivo pueden escalar de manera efectiva cuando se incorpora un nuevo atributo a optimizar. Al integrar simultáneamente similitud, género y raza, el estudio evidencia que el enfoque evolutivo mantiene su capacidad de convergencia y preservación de diversidad.

3.2.3. Estado del arte de los sistemas de reconocimiento facial

Serengil y Ozpinar (2024) llevaron a cabo un estudio exhaustivo sobre las diferentes combinaciones de las etapas del proceso de reconocimiento facial para evaluar su usabilidad y desempeño. El trabajo se enfocó en el proceso de reconocimiento facial, dividido en cuatro etapas aceptadas por investigadores en el área. Las etapas son detección, alineación, representación y verificación. Los autores evaluaron nueve modelos de reconocimiento facial de estado del arte, seis modelos de detección de rostros avanzados, tres funciones de distancia y dos modos de alineación. La metodología implicó experimentos con diferentes combinaciones de las etapas para identificar las configuraciones óptimas de los sistemas de reconocimiento facial. Los experimentos se realizaron utilizando la biblioteca DeepFace que implementa los modelos de reconocimiento, los modelos de detección y permite aplicar métricas y modos de alineación para construir diferentes sistemas de reconocimiento facial. El conjunto de datos utilizado en el estudio fue LFW. Los resultados del estudio demostraron que algunos modelos de reconocimiento facial, como FaceNet-512d, VGG-Face y Dlib, alcanzaron o superaron la precisión calculada para los humanos en ciertas configuraciones. FaceNet-512d, por ejemplo, logró una precisión del 98,4 %, VGG-Face alcanzó el 96,7 % y Dlib superó el 96,8 %. Sin embargo, otros modelos como OpenFace, Facebook DeepFace y DeepId mostraron un desempeño inferior al esperado.

Este artículo aporta un marco comparativo sólido para comprender cómo distintas configuraciones de detección, alineación, representación y verificación afectan el desempeño global de un sistema de reconocimiento facial. Su utilidad para este proyecto radica en disponer de una referencia empírica que sustenta la selección de los sistemas de reconocimiento facial basados en VGG-Face y Dlib.

3.2.4. Sesgo en los conjuntos de datos de imágenes faciales

El estudio de Kärkkäinen y Joo (2021) se centró en la problemática del fuerte sesgo racial y de género presente en los conjuntos de datos públicos de imágenes faciales que son utilizados para el desarrollo de sistemas de reconocimiento facial. Los autores señalaron que los datos existentes estaban fuertemente sesgados hacia los rostros de raza blanca (alrededor del 80 %), lo que resultaba en una precisión de clasificación inconsistente entre grupos demográficos.

El objetivo de la investigación de Kärkkäinen y Joo (2021) fue demostrar que un conjunto de datos balanceado mejora la capacidad de generalización de los modelos, logrando una precisión independiente de la raza y el género. Los autores propusieron FairFace, un conjunto de datos novedoso que contiene 108,501 imágenes balanceadas en función de raza, género y edad. FairFace definió siete grupos raciales (blanca, negra, latina, asiático oriental, asiático sur-oriental, india y medio-oriental), siendo el primer conjunto de datos a gran escala en incluir las razas latina y medio-oriental y en diferenciar entre asiático sur-oriental y asiático oriental.

Los modelos entrenados con FairFace fueron evaluados empíricamente frente a modelos basados en conjuntos de datos sesgados (como UTKFace o LFWA+). En su evaluación, demostraron que un modelo entrenado con el conjunto de datos FairFace alcanzó una precisión de clasificación significativamente más consistente entre distintas razas y géneros. Los resultados subrayaron que la diversidad del conjunto de entrenamiento fue un factor crucial para mejorar la generalización del modelo, estableciendo al conjunto de datos FairFace como un recurso valioso para el entrenamiento de nuevos modelos más equitativos y para la evaluación del sesgo en clasificadores existentes.

El estudio de [Kärkkäinen y Joo \(2021\)](#) proporciona un fundamento empírico para abordar el sesgo racial en los sistemas de clasificación de atributos, argumentando que la diversidad del conjunto de entrenamiento es determinante para mejorar la equidad racial. Su utilidad para este proyecto es argumentar el uso de FairFace como conjunto de datos menos sesgado.

3.2.5. Resumen de los artículos relevados

La [Tabla 3.1](#) presenta año, autores, nombre y concepto clave de los artículos relevados.

<i>Año</i>	<i>Autores</i>	<i>Título</i>	<i>Concepto clave</i>
2021	Machín, Nesmachnow y Toutouh	Evolutionary latent space search for driving human portrait generation	Búsqueda evolutiva monoobjetivo en el espacio latente de StyleGAN2 guiada por FaceNet
2022	Machín, Nesmachnow y Toutouh	Multi-target evolutionary latent space search of a generative adversarial network for human face generation	Búsqueda evolutiva para aproximar simultáneamente dos rostros objetivo en el espacio latente de StyleGAN2
2023	Correa, Mig- naco, Rey, Machín, Nesmachnow y Toutouh	Multiobjective evolutionary search of the latent space of Generative Adversarial Networks for human face generation	MOEA con StyleGAN3 que combina similitud facial y atributo racial utilizando VGG-Face
2025	Nesmachnow y Moreno	Three-objective evolutionary search of the latent space of Generative Adversarial Networks for human face generation	Búsqueda evolutiva con tres objetivos en StyleGAN3 considerando simultáneamente similitud, raza y género
2024	Serengil y Ozpinar	A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules	Punto de referencia de sistemas de reconocimiento facial y usabilidad de módulos en DeepFace
2021	Karkkainen y Joo,	FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation	Conjunto de datos balanceado para medición y mitigación del sesgo racial y de género en reconocimiento facial

Tabla 3.1: Recopilación de trabajos relevados.

Capítulo 4

Arquitectura de la solución

Este capítulo presenta la arquitectura de la solución desarrollada. El pipeline integra un algoritmo evolutivo con módulos de generación, reconocimiento y clasificación racial. Se describen los componentes que conforman el sistema, su función y la forma en que se integran para optimizar la generación de rostros sintéticos que se asemejan a un rostro objetivo y cumplen con un atributo de raza predefinido.

4.1. Estructura general de la solución

El sistema se compone de cuatro módulos principales: un algoritmo evolutivo responsable de la optimización multiobjetivo, un módulo de generación de rostros sintéticos basado en una red generativa antagónica, un módulo de reconocimiento facial encargado de evaluar la similitud entre las imágenes generadas y un rostro objetivo, y un módulo de clasificación racial que determina la similitud de cada imagen a una raza predefinida.

El algoritmo evolutivo dirige la búsqueda en el espacio latente de vectores que representan posibles soluciones al problema. En cada iteración, los vectores se procesan en el módulo de generación para producir imágenes faciales sintéticas. Las imágenes son procesadas en los módulos de reconocimiento y de clasificación de raza, que evalúan la similitud con el rostro objetivo y el grado de correspondencia con la categoría racial definida. Los resultados de estas evaluaciones constituyen las medidas utilizadas para calcular los valores de fitness de cada individuo en función de los objetivos establecidos.

Los valores de fitness determinan la aptitud relativa de cada individuo dentro de la población. Luego, el algoritmo aplica operadores de selección, cruzamiento y mutación para conformar una nueva población de vectores latentes. El ciclo evolutivo se ejecuta durante un número fijo de generaciones. Al finalizar la ejecución del algoritmo, se obtiene un conjunto de soluciones que representa un equilibrio entre los objetivos de similitud facial y correspondencia racial.

4.2. Algoritmo evolutivo

Esta sección presenta la estructura interna del algoritmo evolutivo. Se describe la representación de los individuos, la función de fitness, los operadores evolutivos y los parámetros generales que definen la configuración del algoritmo.

4.2.1. Representación de los individuos

En el contexto de este trabajo, cada individuo de la población representa la entrada al módulo de generación de rostros. Los individuos se codifican mediante un vector latente de 512 valores reales, correspondiente a un punto en el espacio latente de la red generativa antagónica. Cada componente del vector actúa como un alelo y el conjunto completo constituye el genotipo del individuo. Al ser procesado por la GAN, el vector latente se transforma en una imagen facial sintética única que representa el fenotipo asociado dentro del espacio de soluciones del problema.

4.2.2. Función de evaluación

El algoritmo define dos funciones objetivo: similitud facial (F_d) y correspondencia de raza (F_r). El primer objetivo evalúa el grado de similitud entre el rostro generado y el rostro objetivo. Para ello, se emplean dos sistemas de reconocimiento facial basados en los modelos VGG-Face y Dlib respectivamente. Los sistemas de reconocimiento permiten extraer los embeddings de ambas imágenes y calcular la distancia coseno correspondiente a cada par de imágenes.

Dado que los umbrales de similitud difieren significativamente entre los modelos, con valores experimentales de 0.71 para VGG-Face y 0.095 para Dlib, se aplica una ponderación para equilibrar su contribución en la función objetivo. El procedimiento de selección para el valor de ponderación se presenta en la [Subsección 6.1.1](#). La función de similitud combinada se calcula mediante la [Ecuación 4.1](#).

$$F_d = \frac{d_{\text{VGG-Face}} + 10 \cdot d_{\text{Dlib}}}{2} \quad (4.1)$$

El segundo objetivo busca optimizar la correspondencia racial del rostro generado con una raza objetivo. Para ello, se utiliza el modelo FairFace que produce un vector de siete componentes asociado a las probabilidades de pertenencia a una raza. El vector resultante se normaliza y se compara con un vector unitario que representa la raza objetivo. La función de distancia de raza se obtiene a través de la distancia euclidiana entre ambos vectores presentada en la [Ecuación 4.2](#).

$$F_r = |\mathbf{r}_{\text{target}} - \mathbf{r}_{\text{scaled}}| \quad (4.2)$$

El algoritmo evolutivo busca optimizar simultáneamente ambas funciones (F_d, F_r) , con el objetivo de generar individuos que presenten un alto nivel de similitud facial sin depender de un único sistema de reconocimiento y correspondan con la raza objetivo.

4.2.3. Operadores evolutivos

Esta sección describe los operadores evolutivos utilizados en el algoritmo. La selección de los operadores sigue el patrón adoptado en trabajos previos (Correa y cols., 2023).

Selección de padres

La selección se realiza mediante un esquema de torneo binario, utilizado para conformar el conjunto de padres de cada generación. En cada torneo se eligen aleatoriamente dos individuos del conjunto de la población (M. Mitchell, 1999). Entre los individuos seleccionados, el mejor se determina en función de la dominancia de Pareto: un individuo domina a otro si presenta valores iguales o inferiores en todos los objetivos y es estrictamente mejor en al menos uno de ellos. Cuando los individuos seleccionados se encuentran en el mismo nivel de dominancia, la decisión se establece según la distancia de crowding, que permite priorizar a los individuos ubicados en regiones menos densas del frente de Pareto. Este criterio contribuye a preservar la diversidad de la población. Finalmente, el proceso se repite hasta completar el conjunto de padres que participará en las etapas de cruzamiento y mutación (Deb y cols., 2002).

Cruzamiento

El operador de cruzamiento utilizado corresponde al método Blend- α . Este operador genera descendientes mediante una recombinación lineal de los valores de los alelos de dos individuos seleccionados (Eshelman y Schaffer, 1993). Para cada par de padres $A = (a_1, a_2, \dots, a_n)$ y $B = (b_1, b_2, \dots, b_n)$, el valor de cada alelo del descendiente h_i se obtiene seleccionando un punto dentro del intervalo definido por los valores de los padres y el parámetro α como lo expresa la Ecuación 4.3. El valor de cada alelo se selecciona aleatoriamente y con distribución uniforme. En este proyecto se establece $\alpha = 0,2$ siguiendo con los lineamientos del trabajo de Correa y cols. (2023).

$$h_i \in [\min(a_i, b_i) - \alpha|a_i - b_i|, \max(a_i, b_i) + \alpha|a_i - b_i|]. \quad (4.3)$$

Mutación

Este proyecto utiliza mutación gaussiana que primero decide para cada alelo si corresponde aplicar la mutación, dependiendo de una probabilidad definida. Si corresponde, se modifica el valor del alelo sumando un valor aleatorio extraído de una distribución normal con media $\mu = 0$ y desviación estándar $\sigma = 1$.

La probabilidad de mutación de los alelos se establece en $100/512$ y la probabilidad de que a un individuo se le aplique la mutación se fija en $p_m = 0,001$.

NSGA-II

El método NSGA-II, propuesto por (Deb y cols., 2002), combina elitismo y preservación de diversidad para mantener un conjunto de soluciones no dominadas de alta calidad. En cada generación, el algoritmo combina la población actual con los descendientes y organiza los individuos en frentes de no dominancia. Los individuos del primer frente (F_1) representan las soluciones no dominadas del conjunto combinado y los frentes subsiguientes ($F_2, F_3, \dots$) agrupan las soluciones dominadas en diferentes rangos.

Para mantener la diversidad, cada individuo dentro de un frente recibe un valor de distancia de crowding, que cuantifica la densidad de individuos vecinos en el espacio objetivo. La nueva población se construye de forma secuencial agregando frentes hasta alcanzar el tamaño establecido; si el último frente excede dicho límite, se priorizan los individuos con mayor distancia de crowding.

4.2.4. Parámetros generales del algoritmo

El algoritmo evolutivo opera sobre una población de 80 individuos, que evolucionan a lo largo de 100 generaciones mediante los operadores de selección, cruzamiento y mutación definidos. La población inicial se genera de forma aleatoria, asignando a cada dimensión del vector del espacio latente de la GAN un valor seleccionado de una distribución normal estándar. Los parámetros empleados en la configuración del algoritmo se resumen en la [Tabla 4.1](#).

<i>Parámetro</i>	<i>Valor</i>
Tamaño de la población (N)	80
Número de generaciones (G)	100
Probabilidad de cruzamiento (p_c)	0.9
Parámetro del operador BLX- α (α)	0.2
Probabilidad de mutación (p_m)	0.001
Tipo de mutación	Gaussiana ($\mu = 0, \sigma = 1$)
Criterio de selección y ordenamiento	NSGA-II

Tabla 4.1: Parámetros generales del algoritmo evolutivo.

4.3. Generación de rostros sintéticos

Esta sección describe el módulo encargado de la generación de rostros sintéticos. Su función es producir imágenes faciales sintéticas realistas a partir de vectores latentes de alta dimensionalidad que representan las posibles soluciones evaluadas por el algoritmo evolutivo.

En concordancia con la implementación de trabajos anteriores (Correa y cols., 2023), se utiliza la biblioteca StyleGAN3 desarrollada por Nvidia. StyleGAN3 define una red generativa antagónica preentrenada sobre el conjunto de datos Flickr-Faces-HQ (Karras y cols., 2021). El modelo utilizado corresponde a la versión `stylegan3-r-ffhq-1024×1024`. Cada vector de entrada ($\mathbf{z} \in \mathbb{R}^{512}$) se decodifica en una imagen única representada en formato RGB de 1024×1024 que resulta en una tupla de dimensiones (3, 1024, 1024).

La principal innovación de StyleGAN3 respecto a su antecesor, StyleGAN2 (Karras, Laine, y Aila, 2019), es su tratamiento del espacio de generación. En versiones anteriores, los detalles del rostro dependían directamente de las coordenadas de los píxeles, lo que generaba inconsistencias al desplazar o rotar las imágenes: rasgos como ojos, nariz o lunares no se movían de manera coherente con el resto del rostro. StyleGAN3 soluciona este problema al redefinir las convoluciones en un dominio continuo, eliminando la dependencia de las coordenadas absolutas y reduciendo el fenómeno de aliasing. Gracias a las actualizaciones realizadas, las características faciales se trasladan y rotan de forma conjunta con el rostro, preservando la coherencia espacial y visual de las imágenes generadas.

Las propiedades de StyleGAN3 hacen que sea adecuada para su integración con un algoritmo evolutivo ya que el espacio latente continuo responde de manera estable a las operaciones de cruzamiento y mutación, lo que permite una exploración fluida y controlada del espacio de soluciones y la generación imágenes faciales sintéticas de alta resolución y coherencia visual.

4.4. Reconocimiento facial

Esta sección describe la arquitectura del módulo de reconocimiento facial y su integración dentro del pipeline. Se detallan los componentes que conforman el módulo, las bibliotecas utilizadas y los modelos incorporados. Además, se presenta el flujo de procesamiento completo, desde la recepción de las imágenes generadas hasta la obtención de las medidas de similitud empleadas en la función de evaluación.

4.4.1. Estructura general del sistema de reconocimiento

El flujo de procesamiento comienza con la imagen generada por StyleGAN3, que se recibe en formato RGB. Dado que la biblioteca DeepFace opera internamente en formato BGR, se ajusta el orden de los canales de color para asegurar compatibilidad entre los módulos. A continuación, el detector facial identifica el rostro y genera un recorte normalizado. La imagen resultante se transforma a la resolución requerida y se alinea antes de ser procesada por el modelo de reconocimiento. El flujo se realiza en procesos paralelos e independientes para los dos sistemas de reconocimiento utilizados. Finalmente, los vectores de características obtenidos se emplean para calcular las funciones objetivo descritas en la Subsección 4.2.2.

4.4.2. Biblioteca y modelos utilizados

En la solución propuesta se integraron dos sistemas independientes de reconocimiento facial, cada uno compuesto por su propio detector y modelo de reconocimiento. El primero emplea MtCnn como detector y VGG-Face como reconocedor. El segundo utiliza la biblioteca Dlib, que provee el detector y el modelo de reconocimiento. Ambos sistemas se integran mediante la biblioteca DeepFace, en su versión 0.0.94, que actúa como un contenedor que facilita la interacción entre distintos modelos de detección y reconocimiento facial.

VGG-Face

El modelo VGG-Face está basado en la arquitectura VGG-16, adaptada por el grupo Visual Geometry Group de la Universidad de Oxford, para reconocimiento facial (Parkhi y cols., 2015). El modelo consta de una red neuronal de 13 capas convolucionales con filtros de 3×3 , organizadas en cinco bloques secuenciales, cada uno seguido de una capa de max-pooling. A continuación se incorporan tres capas totalmente conectadas. La entrada al modelo son imágenes BGR alineadas de tamaño 224×224 píxeles y la salida es un vector de 2622 dimensiones. Para extraer el vector de características se elimina la capa de clasificación softmax y se utiliza el vector de activación generado en la segunda capa totalmente conectada que resulta en un embedding de 4096 dimensiones.

VGG-Face fue entrenado sobre un conjunto de datos elaborado también por el grupo Visual Geometry Group, que contiene aproximadamente 2.6 millones de imágenes. En tareas de verificación sobre el conjunto LFW, el modelo alcanzó una precisión del 98,95 %.

MtCnn

El detector Multi-task Cascaded Convolutional Networks (MtCnn) propuesto por K. Zhang y cols. (2016) constituye el componente responsable de la detección y alineación facial en el sistema de reconocimiento asociado al modelo VGG-Face. Su arquitectura está compuesta por tres redes neuronales convolucionales conectadas: P-Net, R-Net y O-Net, que trabajan de manera jerárquica para localizar con precisión las regiones faciales. P-Net realiza una detección preliminar de candidatos a rostro a partir de un muestreo piramidal de la imagen. R-Net realiza la detección eliminando falsos positivos y ajustando los límites de las cajas de recorte. Finalmente, O-Net produce las coordenadas finales del rostro y los puntos de referencia faciales, que se utilizan para la alineación, como se muestra en la Figura 4.1. El proceso de detección en cascada permite obtener recortes normalizados y alineados, condición necesaria para la extracción coherente de embeddings en las etapas posteriores del sistema de reconocimiento.

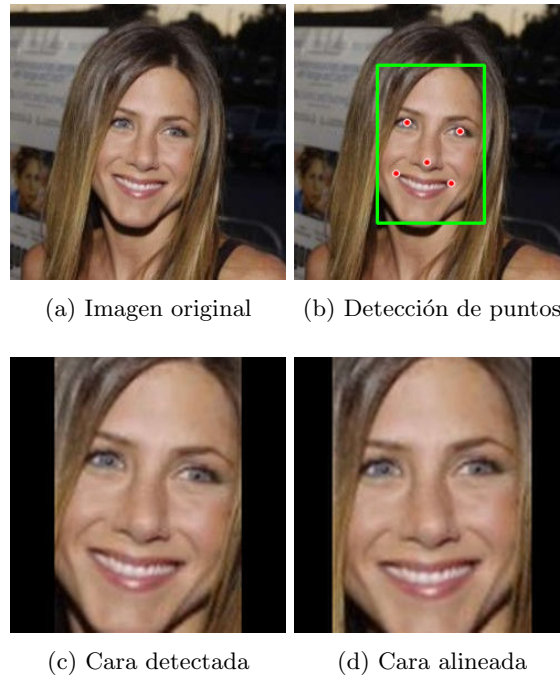


Figura 4.1: Funcionamiento del detector de la biblioteca MtCnn.

Dlib

El modelo de reconocimiento facial implementado en la biblioteca Dlib se inspira en la arquitectura ResNet-34 propuesta por (He y cols., 2016). En particular, corresponde a una versión reducida de 29 capas convolucionales y se reduce a la mitad el número de filtros por bloque (King, 2015).

La red neuronal toma como entrada una imagen facial alineada y normalizada de tamaño 150×150 píxeles en formato BGR, obtenida tras la etapa de detección y alineamiento de puntos faciales. La salida del modelo es un embedding de 128 dimensiones que codifica las características del rostro.

El modelo fue entrenado sobre los conjuntos de datos FaceScrub y VGG-Face2, que en conjunto contienen más de tres millones de imágenes de rostros. En los experimentos realizados sobre el conjunto de validación LFW, el modelo de Dlib alcanzó una precisión del 99.38 %, superando el 97.53 % obtenido por los humanos sobre el mismo conjunto de datos (Taigman y cols., 2014).

Detector Dlib

El detector Dlib se construye mediante una función que permite estimar la posición de cinco puntos clave del rostro. El proceso de detección devuelve una lista de regiones faciales normalizadas, cada una representada por las coordenadas del área detectada y los puntos de referencia correspondientes.

Una vez identificada la región del rostro, el modelo aplica un proceso de alineación geométrica basado en los puntos oculares, garantizando que el rostro quede centrado y orientado antes de su análisis por el reconocedor (King, 2015). Este procedimiento mejora la precisión de los embeddings generados y reduce las variaciones causadas por la pose o la inclinación de la cabeza. El procedimiento de detección y alineación se ilustra en la Figura 4.2.

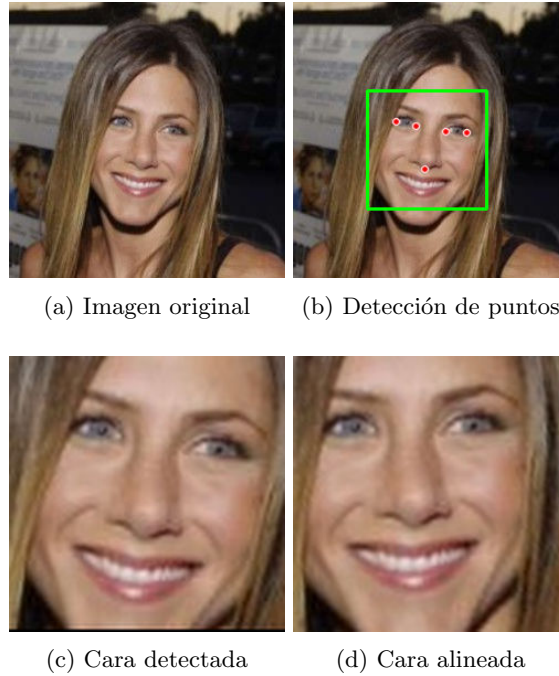


Figura 4.2: Funcionamiento del detector de la biblioteca Dlib.

4.5. Análisis de raza

Esta sección presenta el módulo de clasificación de raza implementado. Se describe el modelo utilizado para la clasificación de atributos de raza y las adaptaciones realizadas para su utilización dentro del sistema.

4.5.1. FairFace

El modelo de la biblioteca FairFace está basado en una red neuronal convolucional profunda de 34 capas, pre entrenada sobre un conjunto de datos diverso y balanceado con respecto a raza, género y edad (Kärkkäinen y Joo, 2021). El modelo FairFace genera un vector de salida de 18 componentes. Las primeras siete componentes corresponden a las categorías raciales: blanca, negra, latina, asiático oriental, asiático sur-oriental, india y medio oriental y las restantes se

asocian a atributos de género y edad que no fueron considerados. La salida se normaliza mediante una función softmax aplicada a las siete primeras componentes, lo que genera una distribución de probabilidad asociada a las siete categorías raciales. El vector probabilístico se compara con el vector unitario correspondiente a la raza objetivo y es esta comparación de distancias la que constituye la función objetivo F_r definida en la [Subsección 4.2.2](#).

Para optimizar la integración del modelo, se modificó la etapa de detección y alineación de la imagen que proporciona la biblioteca FairFace por la detección realizada con Dlib. De este modo, el clasificador recibe directamente la imagen procesada con el detector Dlib. Esta modificación reduce significativamente el tiempo de ejecución y el uso de memoria sin afectar la precisión del modelo.

4.6. Modificaciones implementadas para optimizar la eficiencia computacional

La arquitectura propuesta incorpora una serie de modificaciones orientadas a reducir el tiempo de cómputo y mejorar la eficiencia general del sistema durante la evaluación de cada individuo del algoritmo evolutivo.

La primera modificación se relaciona con una actualización del mecanismo de obtención de embeddings de la biblioteca DeepFace. En lugar de procesar cada imagen de manera individual, se emplea el nuevo esquema de procesamiento por lotes, lo que permite aprovechar de manera más eficiente la capacidad de paralelización interna de cada modelo de reconocimiento y reducir el tiempo de ejecución.

La segunda modificación consiste en la ejecución paralela de los dos reconocedores utilizados. Se crean dos procesos independientes que operan de manera simultánea sobre el mismo conjunto de imágenes generadas. La modificación permite aprovechar mejor los recursos disponibles y reducir el tiempo total de la evaluación de cada individuo.

Finalmente, se incorpora una tercera modificación vinculada al cálculo del atributo racial. Tal como se menciona en la [Subsección 4.5.1](#), el sistema aprovecha la imagen detectada previamente por Dlib como entrada al clasificador de raza, evitando la necesidad de volver a procesar la detección y realizar recortes de rostro adicionales. La reutilización de la representación ya normalizada no solo elimina operaciones repetidas, sino que también contribuye a una reducción significativa del consumo de memoria y del tiempo de ejecución asociado a esta etapa. Los resultados de reducción de tiempo de ejecución se presentan en la [Tabla 6.2](#).

4.7. Funcionamiento del pipeline completo

El funcionamiento general del pipeline se describe con una secuencia de pasos que aplica a todos los individuos de la población en cada generación.

1. Para cada individuo, se toma su vector latente como entrada para el generador de imágenes de StyleGAN3.
2. La imagen se envía a los dos sistemas de reconocimiento para ser detectada, alineada y generar los embeddings.
3. Se calcula la similitud facial con el rostro objetivo utilizando cada embedding generado en el paso anterior.
4. Se calcula la comparación con la raza objetivo a partir del vector de probabilidades de FairFace y el vector unitario correspondiente.
5. Los resultados de similitud facial y de raza determinan los valores de aptitud de cada individuo dentro de la población.
6. El algoritmo evolutivo aplica los operadores de selección, cruzamiento y mutación para generar la nueva población.
7. Se repite el ciclo durante un número fijo de generaciones, refinando progresivamente la población hasta obtener un conjunto de individuos que representa el equilibrio entre los dos objetivos.

La [Figura 4.3](#) ilustra la estructura general del sistema y la interacción entre los módulos que lo conforman.

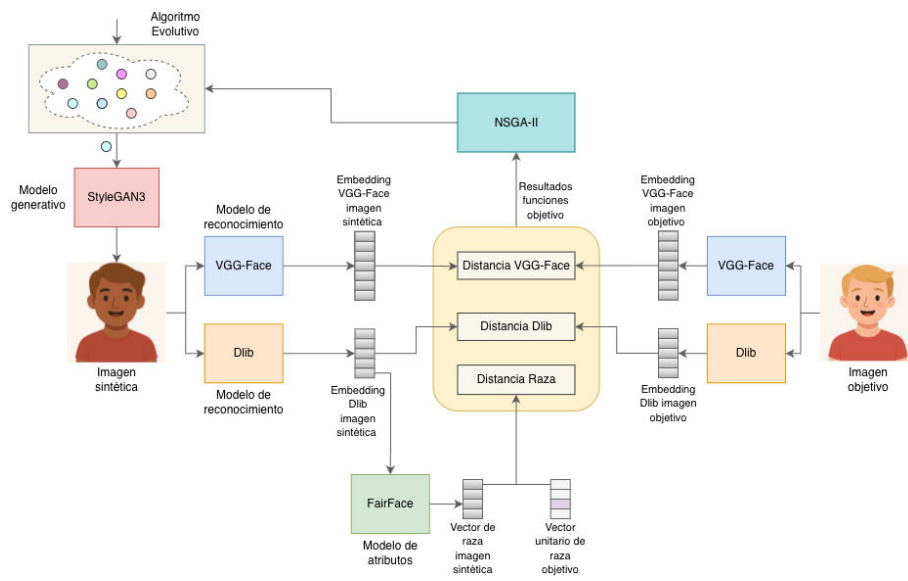


Figura 4.3: Diagrama del funcionamiento del pipeline.

Capítulo 5

Metodología

Este capítulo describe la metodología adoptada para el desarrollo y evaluación del algoritmo propuesto. Se detallan los procedimientos seguidos para la selección de modelos de detección y reconocimiento facial, la elección del clasificador de raza y la evaluación del algoritmo evolutivo.

5.1. Metodología para selección de modelos de detección y reconocimiento

Esta sección describe la metodología utilizada para la selección de los modelos de detección y reconocimiento facial empleados en el pipeline. El proceso de selección se fundamentó en la revisión y análisis de distintos enfoques reportados en el estudio de referencia de [Serengil y Ozpinar \(2024\)](#), que presenta un punto de comparación de diferentes configuraciones de detección, alineamiento, representación y verificación facial utilizando modelos de estado del arte.

De acuerdo con los lineamientos del proyecto, se seleccionaron dos sistemas de reconocimiento facial para reducir el acoplamiento del pipeline generado por una única implementación de reconocimiento. Para el primer sistema de reconocimiento, se seleccionó el modelo de reconocimiento VGG-Face junto con el detector facial de MtCnn continuando con el trabajo previo de [Correa y cols. \(2023\)](#).

El segundo sistema seleccionado fue uno propuesto por la biblioteca Dlib, el cual fue elegido con base en los resultados del punto de comparación, donde mostró un rendimiento competitivo frente a sistemas de mayor complejidad. En los experimentos realizados por [Serengil y Ozpinar \(2024\)](#), el sistema de Dlib alcanzó sus mejores resultados cuando se combinó con su propio detector facial, que logró precisiones del 96.4 % con distancia coseno. Por otro lado, Dlib ofrece ventajas significativas en términos de eficiencia computacional, lo que resulta especialmente adecuado para su integración en un sistema basado en un proceso evolutivo que requiere evaluar múltiples generaciones y un volumen elevado de individuos, cada uno representado por una imagen facial.

5.2. Metodología para el análisis de los sistemas de reconocimiento

Esta sección presenta la metodología utilizada para los experimentos sobre los reconocedores faciales seleccionados. Cada experimento se realizó de forma independiente tomando como base el conjunto de datos LFW.

5.2.1. Metodología para la ponderación de distancias entre sistemas de reconocimiento facial

El objetivo de esta etapa fue establecer un factor de ponderación que permitiera comparar de manera coherente las distancias generadas por los sistemas basados en los modelos VGG-Face y Dlib. Ambos modelos producen vectores de distinta dimensionalidad y fueron entrenados con criterios diferentes, por lo que sus distancias coseno no son directamente comparables.

Se utilizó un listado de 6000 pares de imágenes de LFW divididos equitativamente entre pares correspondientes a la misma persona y pares de distintas personas. Para cada par se procesaron las imágenes con Dlib y con VGG-Face, utilizando para cada modelo el detector seleccionado y extrayendo los embeddings correspondientes. A partir de los embeddings de cada reconocedor, se calculó la distancia coseno entre las dos imágenes, generando así dos conjuntos de distancias: uno para Dlib y otro para VGG-Face.

Posteriormente, se emparejaron ambos conjuntos de distancias para obtener, por cada par del listado, la proporción entre la distancia generada por VGG-Face y la generada por Dlib. Esto permitió estimar la relación entre las distancias. Para cada categoría (misma persona y diferentes personas) se calcularon los valores media, mediana y desviación estándar de las proporciones. Este procedimiento sirvió como base para determinar el factor de ponderación utilizado en la primera función objetivo del algoritmo propuesto.

5.2.2. Metodología para el cálculo experimental de los umbrales

El objetivo de esta etapa fue determinar un umbral de decisión adecuado para distinguir entre pares de imágenes que corresponden a la misma persona y pares pertenecientes a personas distintas. Para este experimento se utilizaron las distancias calculadas en la [Subsección 5.2.1](#).

El umbral se define como la distancia máxima aceptable entre dos representaciones vectoriales para decidir que ambas imágenes corresponden al mismo individuo. Si la distancia se encuentra por encima del umbral, se decide que pertenecen a personas distintas. Un umbral muy bajo corre el riesgo de clasificar imágenes de la misma persona como personas distintas y un umbral muy alto puede clasificar personas distintas como la misma persona.

Para obtener una estimación cuantitativa, se entrenó un árbol de decisión construido mediante el algoritmo ID3. Se utilizó la distancia como única característica y la etiqueta de clase binaria (misma persona/distinta persona) como variable objetivo.

Como segundo método, se calculó la estimación de las distribuciones de distancias mediante el método de estimación de la densidad de Kernel KDE. Se generaron curvas KDE independientes para ambas clases, lo que permitió visualizar el solapamiento entre las distribuciones. El umbral se interpreta como el punto de corte entre ambas curvas.

Finalmente, se evaluó el desempeño del clasificador obtenido por los métodos del árbol de decisión y KDE mediante matrices de confusión sobre un conjunto de validación. Estas matrices permitieron medir directamente la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos asociada al umbral obtenido.

5.3. Metodología para la sustitución del reconocedor de raza

Esta sección presenta el procedimiento seguido para seleccionar el reconocedor de raza del pipeline. Se describen las razones que motivan el reemplazo del clasificador DeepFace, la evaluación del modelo FairFace y su integración con la imagen ya procesada por Dlib. Se presenta la metodología de comparación de FairFace con DeepFace. Finalmente, se describe la metodología para determinar un umbral de aceptación de clasificación racial.

5.3.1. Metodología para la selección del nuevo reconocedor de raza

En trabajos previos, la biblioteca DeepFace ha sido empleada como clasificador de atributos raciales. Sin embargo, [Correa y cols. \(2023\)](#) evidenciaron que DeepFace presentó sesgos significativos, fundamentalmente asociados a la alta proporción de rostros de raza blanca en los conjuntos utilizados para su entrenamiento. La falta de balance deriva en un comportamiento desigual entre categorías raciales y en una disminución notable de la precisión en grupos subrepresentados.

Con el propósito de mitigar los sesgos y mejorar la calidad de la clasificación racial dentro del pipeline propuesto, se evaluaron alternativas al clasificador DeepFace. Se experimentó con el modelo FairFace ([Kärkkäinen y Joo, 2021](#)), entrenado sobre más de cien mil imágenes distribuidas de manera uniforme entre siete categorías raciales para reducir el sesgo.

Durante los experimentos preliminares se identificó que los tiempos de ejecución de FairFace resultaron considerablemente elevados. Este comportamiento se debió principalmente al procesamiento previo de la imagen requerido antes de realizar la clasificación. Con el conjunto de validación de FairFace (Kärkkäinen y Joo, 2021) se comparó la precisión y los tiempos de ejecución entre el procesamiento completo de la imagen y el procesamiento de la imagen obtenida con el detector Dlib. La adaptación buscó reducir de manera significativa los tiempos de ejecución, manteniendo al mismo tiempo el nivel de precisión del modelo original.

5.3.2. Metodología para evaluación del clasificador de raza modificado

Para evaluar el desempeño del clasificador FairFace modificado y compararlo con DeepFace, se llevó a cabo un análisis utilizando el conjunto de datos UTKFace (Z. Zhang, Song, y Qi, 2017), un conjunto ampliamente empleado en tareas de clasificación demográfica. UTKFace proporciona imágenes de individuos acompañadas de etiquetas que incluyen atributos de raza, género y edad. En este experimento se consideraron únicamente las etiquetas correspondientes a la categoría racial. Las categorías raciales de UTKFace son asiática, negra, india, blanca y otras, donde la categoría otras incluye las categorías latina y medio oriental.

El procedimiento para la evaluación consistió en procesar cada imagen del conjunto UTKFace con dos clasificadores, el modelo FairFace con la imagen procesada por Dlib y el clasificador DeepFace utilizado en trabajos previos. Para cada imagen, se registró la predicción asignada por ambos modelos y el tiempo de ejecución. Los resultados fueron posteriormente comparados con las etiquetas originales del conjunto. Para el análisis correspondiente a la categoría racial asiática dentro del conjunto UTKFace, se consideraron las predicciones generadas por FairFace para las clases asiático oriental y asiático suroriental. En el caso de la categoría otros, la comparación se realizó considerando las predicciones de FairFace para las razas latina y medio oriental.

Se examinó el número de aciertos en cada una de las categorías raciales presentes en UTKFace y se determinó si los clasificadores presentaban un rendimiento homogéneo entre los diferentes grupos o si manifestaban una tendencia a cometer errores en categorías específicas. Además, se estudiaron las diferencias en los tiempos de ejecución.

5.3.3. Metodología para el cálculo del umbral de FairFace

El clasificador de raza FairFace produce un vector de probabilidades asociado a las siete categorías de raza. Con el fin de decidir si una predicción puede considerarse fiable, es necesario definir un umbral sobre la distancia entre el vector de salida del clasificador y la etiqueta real, de modo que distancias menores al umbral se interpreten como predicciones correctas y distancias superiores como predicciones incorrectas. Un umbral demasiado alto podría admitir como correctas predicciones claramente erróneas y un umbral demasiado bajo descartaría casos que, en la práctica, resultan razonablemente aceptables.

Para el cálculo experimental de este umbral se empleó un conjunto de más de 10 mil imágenes de un conjunto de validación de FairFace etiquetadas con las 7 categorías raciales. Para cada imagen se generó un vector de referencia que representa la raza verdadera mediante una codificación one-hot, asignando el valor 1 a la categoría racial correspondiente y 0 al resto. La discrepancia entre la etiqueta real y la predicción del modelo FairFace se cuantificó mediante la distancia euclidiana entre el vector de referencia y la salida probabilística del clasificador. A partir del conjunto de distancias se aplicaron tests estadísticos para determinar el umbral significativo para aceptar una clasificación de raza.

5.4. Metodología para el análisis de sesgo en la generación y evaluación de rostros

Para evaluar la existencia de sesgo racial en el uso combinado de StyleGAN3 y el clasificador FairFace se generaron 150 mil imágenes con StyleGAN3 a partir de vectores latentes seleccionados al azar. Cada imagen fue clasificada utilizando la versión modificada de FairFace, que asignó el vector de probabilidades raciales. A partir de los resultados, se analizó la distribución de razas para poder identificar patrones introducidos por StyleGAN3 en la síntesis de rostros y por FairFace modificado en la clasificación racial.

5.5. Metodología para la evaluación del algoritmo propuesto

Esta sección define la metodología utilizada para evaluar el algoritmo evolutivo propuesto. Se detallan los criterios adoptados para la selección de rostros, la configuración paramétrica, los operadores evolutivos y los procedimientos estadísticos y de validación empleados para analizar su desempeño.

5.5.1. Elección de rostros

Se seleccionaron las imágenes objetivo con el fin de garantizar una representación equilibrada en términos de género, raza y edad. Para la configuración paramétrica se eligieron cuatro imágenes, de las cuales dos corresponden a rostros de género femenino y dos a género masculino. En términos de raza, se seleccionaron rostros representativos de las razas blanca, negra, asiática e india. La [Figura 5.1](#) presenta las imágenes utilizadas para la configuración paramétrica. La [Tabla 5.1](#) detalla las características nombre, género, raza y edad de cada instancia.



Figura 5.1: Rostros utilizados para la configuración paramétrica. De arriba a abajo y de izquierda a derecha: I1, I2, I3 e I4.

<i>Instancia</i>	<i>Nombre</i>	<i>Género</i>	<i>Raza</i>	<i>Raza objetivo</i>
I1	Dev Patel	Masculino	India	Blanca
I2	Lupita Nyong'o	Femenino	Negra	Asiática oriental
I3	Ana Paula Mazzeo	Femenino	Blanca	India
I4	Hombre asiático generado	Masculino	Asiática	Negra

Tabla 5.1: Características de las imágenes utilizadas en la configuración paramétrica.

Para la etapa de validación se incorporaron nueve imágenes adicionales, distribuidas en cuatro rostros de género femenino y cinco de género masculino. La selección de instancias mantuvo los criterios de diversidad racial, incluyendo representaciones de individuos de raza blanca, negra, asiática, india y latina.

La [Figura 5.2](#) presenta las imágenes utilizadas en la validación y las características demográficas se detallan en la [Tabla 5.2](#).

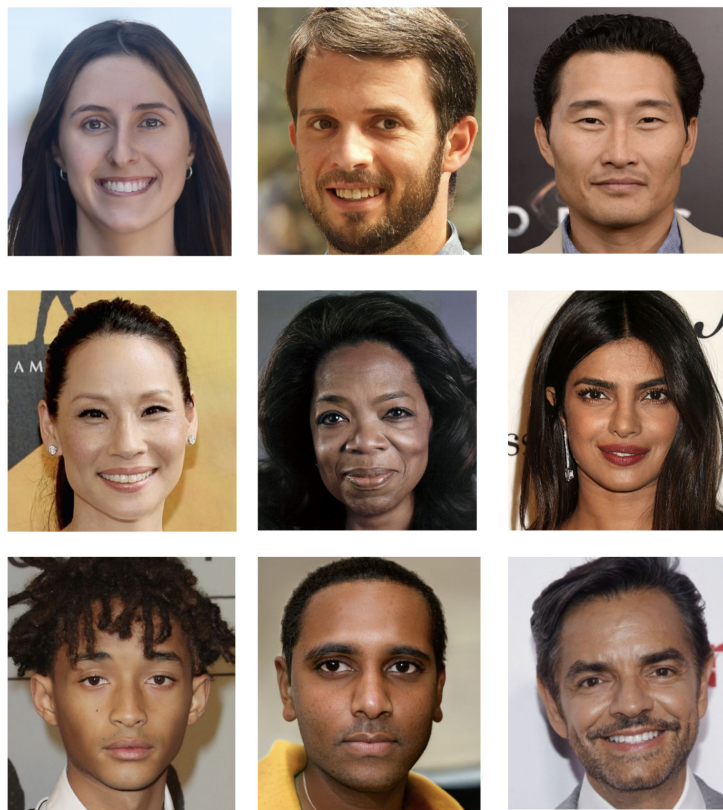


Figura 5.2: Rostros utilizados para la configuración paramétrica. De arriba a abajo y de izquierda a derecha: I5, I6, I7, I8, I9, I10, I11, I12, I13.

<i>Instancia</i>	<i>Nombre</i>	<i>Género</i>	<i>Raza</i>	<i>Raza objetivo</i>
I5	Agustina Mazzeo	Femenino	Blanca	Negra
I6	Hombre blanco generado	Masculino	Blanca	India
I7	Daniel Dae Kim	Masculino	Asiática	India
I8	Lucy Liu	Femenino	Asiática	Latina
I9	Oprah Winfrey	Femenino	Negra	Asiática oriental
I10	Priyanka Chopra	Femenino	India	Negra
I11	Jaden Smith	Masculino	Negra	Blanca
I12	Hombre indio generado	Masculino	India	Blanca
I13	Eugenio Derbez	Masculino	Latina	Asiática oriental

Tabla 5.2: Características de las imágenes utilizadas en la validación del algoritmo.

5.5.2. Métricas utilizadas para la evaluación

Las métricas descritas en esta sección se aplicaron en todas las etapas pertinentes de la evaluación del algoritmo evolutivo.

Hipervolumen relativo

Para permitir la comparación entre configuraciones paramétricas y distintos escenarios experimentales, se utilizó el hipervolumen relativo. La métrica normaliza el hipervolumen obtenido en cada ejecución respecto al hipervolumen de un frente de Pareto de referencia asociado a la instancia analizada.

El frente de referencia se construyó a partir de la unión de todas las soluciones no dominadas generadas por el algoritmo en todas las ejecuciones realizadas para una instancia. Para ello, se recopilieron todas las soluciones no dominadas obtenidas y se aplicó una función de filtrado que identificó el subconjunto de soluciones no dominadas, constituyendo así una aproximación del frente de Pareto para la instancia considerada. A partir del frente combinado se definió el punto de referencia, determinado por los peores valores registrados para cada función objetivo.

Con el hipervolumen calculado en cada ejecución, el punto de referencia y el hipervolumen asociado al frente de referencia, se obtuvo el hipervolumen relativo. Esta métrica permitió comparar de forma directa el rendimiento de las configuraciones evaluadas.

Una vez calculados los valores de hipervolumen relativo, se aplicaron tests de normalidad con el fin de determinar la mejor forma de reportar los resultados. Cuando la distribución no se ajustó a una distribución normal, los valores se presentaron mediante la mediana y el rango intercuartílico. En caso contrario, se utilizaron la media y la desviación estándar.

Test de Kolmogorov-Smirnov

Dado que los algoritmos evolutivos presentan un comportamiento inherentemente no determinista, para cada configuración se realizaron treinta ejecuciones independientes. Resulta necesario aplicar tests de normalidad para seleccionar las métricas estadísticas adecuadas para reportar los resultados.

El test de normalidad de Kolmogorov-Smirnov es un test estadístico que evalúa la hipótesis nula de que los datos provienen de una distribución normal. Para ello, compara la función de distribución empírica de los datos con la distribución normal teórica y calcula la mayor diferencia entre ambas. A partir de esta comparación se obtiene un p - *valor* donde en caso de ser menor al nivel de significancia establecido se rechaza la hipótesis nula. Para la evaluación de los distintos experimentos se adoptó un nivel de significancia de $\alpha = 0,05$.

Test de rangos de Friedman

Una vez obtenidos los valores de hipervolumen relativo para cada configuración en cada instancia, se empleó el test de rangos de Friedman para determinar si existían diferencias estadísticamente significativas en el rendimiento entre las configuraciones evaluadas.

El test de Friedman asigna rangos a los resultados obtenidos por cada configuración para cada instancia y evalúa la hipótesis nula de que todas las configuraciones presentan un comportamiento equivalente y siguen la misma distribución. El nivel de significancia adoptado fue $\alpha = 0,05$. Entonces un p - *valor* inferior a α implica el rechazo de la hipótesis nula.

Cuando la hipótesis nula es rechazada, se aplica un análisis post-hoc para estimar el rango promedio asociado a cada configuración y establecer cuáles configuraciones presentan diferencias significativas entre sí. Dado que el test de rangos de Friedman considera mejores los valores más cercanos a cero, se invirtieron los rangos promedio con el fin de obtener una interpretación coherente con el hipervolumen relativo. La configuración con el menor rango se consideró la mejor configuración de parámetros y fue seleccionada para la validación.

5.5.3. Elección del tamaño de población

Ejecuciones preliminares del algoritmo confirmaron los altos tiempos de ejecución. Se propuso establecer un tamaño de población adecuado para el EA de modo de equilibrar la calidad de las soluciones obtenidas y el costo computacional del algoritmo. Para ello, se evaluaron dos configuraciones: poblaciones de 56 y 80 individuos. Ambas configuraciones se ejecutaron bajo las mismas condiciones de parámetros.

Para cada configuración se llevaron a cabo treinta ejecuciones independientes y se registraron las soluciones no dominadas obtenidas en cada una de ellas. El criterio principal de comparación fue el hipervolumen relativo, métrica que permitió evaluar la dominancia y la diversidad de las soluciones generadas.

5.5.4. Elección del número de generaciones

Dado el elevado costo computacional asociado a la ejecución del algoritmo evolutivo, se diseñó un experimento destinado a determinar una cantidad de generaciones apropiada. El objetivo fue identificar el punto a partir del cual el incremento en la calidad del frente de Pareto se vuelve despreciable, de manera de evitar un número de generaciones innecesariamente elevado.

Se realizaron treinta ejecuciones independientes del algoritmo evolutivo, cada una de ellas configurada con un límite de 200 generaciones. Esta cantidad se seleccionó para obtener un comportamiento suficientemente prolongado del algoritmo y disponer de datos que permitieran analizar la evolución del hipervolumen a lo largo de un amplio número de generaciones.

Para cada ejecución, se recopilaban las soluciones no dominadas obtenidas en la última generación. La unión de estos conjuntos se sometió a un filtrado de dominancia para construir un frente de Pareto de referencia. Sobre el frente de referencia se calculó el hipervolumen de referencia, denotado por HV_{ref} .

Cada diez generaciones se registró el conjunto de soluciones no dominadas generado por el algoritmo. Para cada generación $g \in \{10, 20, 30, \dots, 200\}$ se calculó el hipervolumen asociado a las soluciones no dominadas obtenidas en esa generación para cada una de las ejecuciones, denotado como $HV_i(g)$. A partir de los valores individuales, se obtuvo el hipervolumen promedio para cada generación ($HV(g)$).

Con el fin de analizar la mejora del algoritmo a lo largo de cada ejecución, se evaluó la variación del hipervolumen relativo en intervalos de diez generaciones consecutivas. Para ello, se consideraron pares de generaciones $(g, g+10)$ y se calculó la mejora porcentual mediante la expresión presentada en la [Ecuación 5.1](#). El análisis de la evolución de esta métrica permitió identificar el punto a partir del cual el incremento del hipervolumen se vuelve despreciable, y se estableció la cantidad óptima de generaciones para las ejecuciones posteriores.

$$MP = \frac{HV(g+10) - HV(g)}{HV(g)} \times 100. \quad (5.1)$$

5.5.5. Elección de la configuración paramétrica

Una vez establecida la cantidad adecuada de generaciones y el tamaño de la población, se procedió a determinar la combinación óptima de las probabilidades de cruzamiento y mutación del algoritmo evolutivo. Con este objetivo se definieron como valores candidatos para la probabilidad de cruzamiento $P_c \in \{0,60, 0,75, 0,90\}$ y para la probabilidad de mutación $P_m \in \{0,001, 0,01, 0,10\}$. La combinación de las probabilidades dio lugar a nueve configuraciones posibles que se presentan en la [Tabla 5.3](#).

La evaluación paramétrica se llevó a cabo utilizando las cuatro instancias definidas en la [Tabla 5.1](#). Para cada una de las instancias se realizaron treinta ejecuciones independientes para cada configuración. Como medida de convergencia, se utilizó la métrica de hipervolumen relativo.

Para comparar las distribuciones de hipervolumenes relativos obtenidas, se aplicaron los tests estadísticos definidos en la [Subsección 5.5.2](#). En primer lugar, se realizaron tests de normalidad con el propósito de determinar si los valores seguían una distribución normal. Dado que los tests estadísticos rechazaron la hipótesis de normalidad, se aplicaron métodos no paramétricos basados en rangos para ordenar las configuraciones de acuerdo con su desempeño global sobre las instancias I1, I2, I3 e I4. Este proceso permitió identificar la combinación de probabilidades de mutación y cruzamiento con mejor rendimiento relativo y seleccionar los valores más apropiados para su utilización en el algoritmo evolutivo.

<i>Configuración</i>	P_c	P_m
C1	0.60	0.001
C2	0.60	0.010
C3	0.60	0.100
C4	0.75	0.001
C5	0.75	0.010
C6	0.75	0.100
C7	0.90	0.001
C8	0.90	0.010
C9	0.90	0.100

Tabla 5.3: Configuraciones paramétricas seleccionadas para evaluar el algoritmo evolutivo.

5.5.6. Metodología para la validación

Una vez finalizada la etapa de configuración paramétrica y determinadas la cantidad de generaciones, el tamaño de la población y las probabilidades de aplicación de los operadores evolutivos para el algoritmo, se procedió a realizar treinta ejecuciones independientes para cada una de las instancias seleccionadas en la [Tabla 5.2](#).

Para cada instancia se calcularon las métricas de convergencia y calidad del frente multiobjetivo. En primer lugar, se construyó un frente de Pareto de referencia aproximado con todas las soluciones no dominadas obtenidas en las treinta ejecuciones. Este frente aproximado se utilizó para determinar el hipervolumen de referencia asociado a cada instancia para calcular el hipervolumen relativo de cada ejecución. A partir de los treinta hipervolumenes relativos calculados se obtuvo la media, la desviación estándar, el máximo y el mínimo para evaluar la convergencia en cada instancia. Además, se calculó el spacing que permitió evaluar la distribución de las soluciones no dominadas en el frente, verificando si éstas se encuentran uniformemente espaciadas o si presentan concentración excesiva en determinadas regiones del espacio objetivo.

El análisis de tiempos de ejecución por instancia se realizó registrando el tiempo total empleado en una ejecución del pipeline completo, incluyendo la detección facial con MtCnn y Dlib, la extracción de embeddings con VGG-Face y Dlib y la clasificación racial con FairFace. Cada instancia fue procesada de manera independiente y bajo condiciones homogéneas en el entorno de cómputo de ClusterUY, utilizando una GPU NVIDIA A40 y 20 GB de memoria RAM.

De forma complementaria, se registró el progreso del algoritmo a lo largo de las generaciones. Para un conjunto representativo de instancias se midió la evolución del hipervolumen relativo y la distribución de las soluciones no dominadas en cada generación, lo que permitió apreciar cómo se construye el frente aproximado durante el proceso evolutivo.

Además, se evaluó la tasa de acierto asociada a los sistemas de reconocimiento utilizados en el pipeline. Para cada instancia, se procesaron todas las soluciones no dominadas de las treinta ejecuciones con VGG-Face, Dlib y FairFace, calculando la proporción de imágenes que se encontraban por debajo de los umbrales establecidos en cada caso. Por otro lado, para cada instancia real se seleccionaron cinco fotografías alternativas y se compararon con cinco soluciones representativas y cinco soluciones aleatorias del frente de referencia aproximado. Estas imágenes se evaluaron con VGG-Face y Dlib. Los resultados del análisis se reportaron con mapas de calor para visualizar cómo varían las distancias de similitud cuando la instancia objetivo se modifica ligeramente.

Finalmente, se evaluó la capacidad del algoritmo para engañar a múltiples reconocedores de estado del arte. Se seleccionaron las imágenes del frente de referencia aproximado y se procesaron por sistemas de reconocimiento facial que no formaron parte del pipeline propuesto. Los reconocedores FaceNet ([Schroff y cols., 2015](#)), ArcFace ([Deng y cols., 2022](#)), Facebook DeepFace ([Taigman y cols., 2014](#)) y SFace ([Zhong y cols., 2021](#)) fueron seleccionados con el objetivo de verificar si las soluciones generadas podían engañar a cualquier reconocedor para evaluar la generalización del ataque.

Capítulo 6

Experimentación

Este capítulo presenta la evaluación experimental realizada en este proyecto. Todos los experimentos presentados fueron ejecutados en el Centro Nacional de Supercomputación de Uruguay (Cluster-UY) (Nesmachnow y Iturriaga, 2019).

6.1. Análisis de los modelos de reconocimiento

Esta sección presenta los análisis para la evaluación de los sistemas de reconocimiento seleccionados.

6.1.1. Ponderación de distancias entre modelos de reconocimiento facial

La [Figura 6.1](#) presenta la distribución de las proporciones entre las distancias generadas por VGG-Face y Dlib. Se distingue entre pares correspondientes a la misma persona y pares correspondientes a distintas personas. Los resultados de la evaluación de las proporciones mostraron que la relación entre las distancias obtenidas con ambos sistemas no fue uniforme. Dada la diferencia de distancias, fue necesario definir un factor de ponderación aplicado a las distancias generadas por Dlib.

Se adoptó un factor igual a 10 reportado en la [Tabla 6.1](#). El factor seleccionado se ubica por encima de la media más una desviación estándar y proporciona un margen de seguridad adecuado frente a las proporciones más altas. A su vez el factor permite una comparación equilibrada entre los embeddings de ambos modelos.

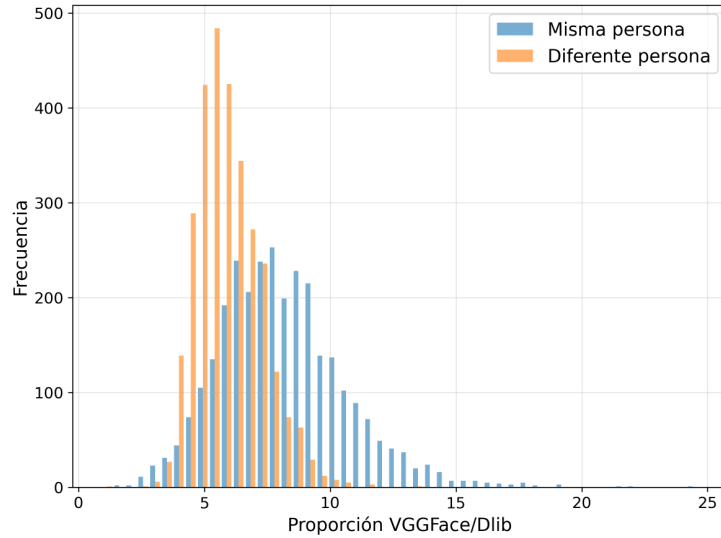


Figura 6.1: Distribución de proporciones para cada reconocedor con imágenes de LFW.

<i>Categoría</i>	<i>Media</i>	<i>Std</i>
Todos los pares	7.05	2.63
Misma persona	8.15	2.96
Diferentes personas	5.94	1.27

Tabla 6.1: Resultados de la evaluación de la proporción entre distancias obtenidas con VGG-Face y Dlib.

6.1.2. Cálculo experimental de los umbrales de reconocimiento

Los valores de distancia obtenidos previamente en [Subsección 5.2.1](#) fueron empleados para estimar los umbrales de decisión correspondientes a los reconocedores VGG-Face y Dlib. Para cada reconocedor, se entrenó un árbol de decisión utilizando como única característica la distancia entre los embeddings. El rendimiento de cada modelo se evaluó mediante las matrices de confusión para el conjunto de validación, presentadas en la [Figura 6.2](#).

Además, se realizó un recorrido sobre un rango continuo de cien valores posibles. Las Figuras [6.3](#) y [6.4](#) muestran la variación del F1-score en función del umbral considerado, destacando el punto que maximiza el desempeño del clasificador. Para Dlib y VGG-Face, el valor obtenido mediante este método resultó muy cercano al umbral sugerido por el árbol de decisión.

Finalmente, la selección del umbral se validó utilizando KDE para las distribuciones de distancia asociadas a ambas clases. La validación, presentada en las Figuras 6.5 y 6.6, evidenció la separación entre las distribuciones y confirmó que el umbral seleccionado se ubicaba en la zona de superposición.

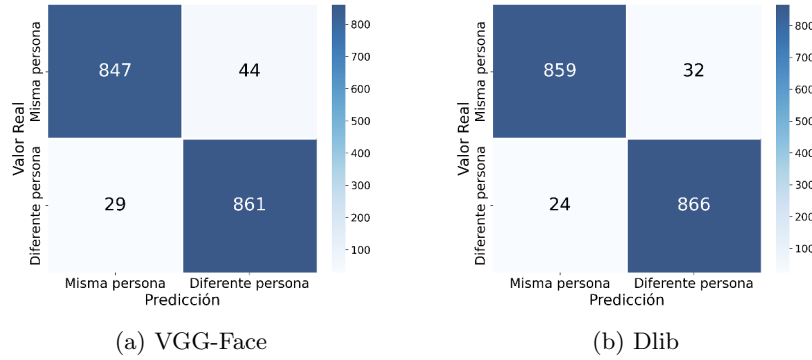


Figura 6.2: Matrices de confusión para los reconocedores utilizando el umbral obtenido por el árbol de decisión.

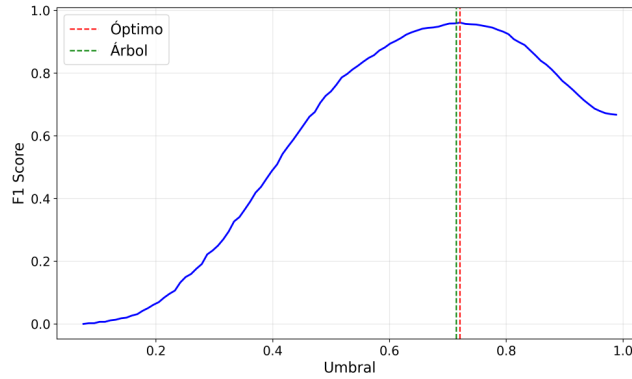


Figura 6.3: Variación del F1-score en función del umbral para VGG-Face.

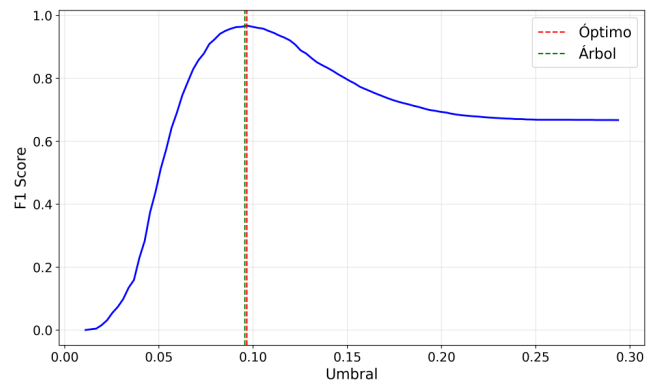


Figura 6.4: Variación del F1-score en función del umbral para Dlib.

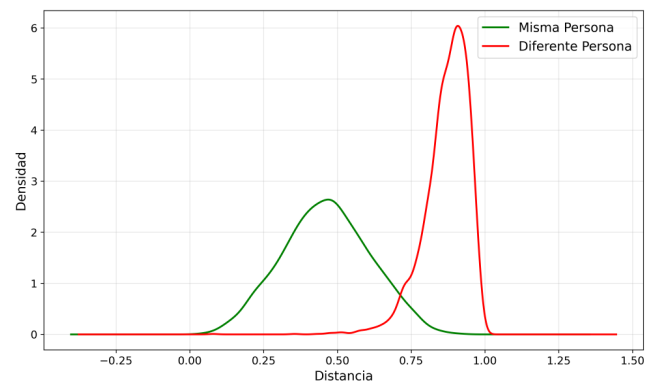


Figura 6.5: Método KDE para validar el umbral para VGG-Face.

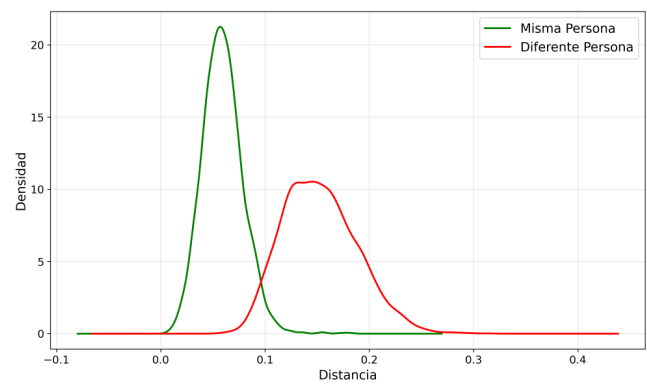


Figura 6.6: Método KDE para validar el umbral para Dlib.

6.2. Evaluación de modelos de clasificación de atributos de raza

Esta sección presenta el análisis experimental realizado para evaluar el desempeño de los modelos utilizados en la clasificación de atributos raciales. En primer lugar, se examinó la eficacia de la modificación aplicada a FairFace usando la imagen procesada por Dlib comparando su precisión y tiempos de ejecución con los del modelo original. Posteriormente, el modelo se comparó con DeepFace empleando el conjunto UTKFace, con el fin de identificar diferencias en precisión, distribución de predicciones de raza y presencia de sesgos.

6.2.1. Evaluación con FairFace

La [Tabla 6.2](#) presenta los resultados de la evaluación realizada con el conjunto de más de diez mil imágenes de validación de FairFace. La tabla muestra la proporción de aciertos obtenida por el modelo original y por la versión modificada. En ambos casos, la precisión se mantuvo mayoritariamente equivalente, lo que indica que el cambio por el detector Dlib no introdujo pérdidas relevantes en la capacidad de clasificación del modelo.

Sin embargo, reemplazar el procesamiento original de la imagen por el uso del detector de Dlib permitió reducir de manera significativa los tiempos de ejecución. A partir de los resultados, se concluyó que la versión modificada de FairFace representó la alternativa más adecuada para su integración dentro del pipeline. Este punto fue importante ya que facilitó la escalabilidad computacional, un factor importante en la integración de otra red neuronal en el algoritmo evolutivo.

<i>Modelo</i>	<i>Aciertos</i>	<i>Tiempo medio (s)</i>
FairFace Original	7408	5.40
FairFace + Dlib	7423	0.61

Tabla 6.2: Aciertos y tiempo medio de ejecución para FairFace y FairFace modificado.

6.2.2. Comparación de DeepFace y FairFace

La [Figura 6.7](#) presenta los resultados de la comparación de las distribuciones reales de razas presentes en UTKFace con aquellas inferidas por ambos clasificadores. La figura evidencia que DeepFace tendió a distorsionar la distribución, particularmente al clasificar más imágenes como personas asiáticas y latinas. FairFace logró una representación sensiblemente más próxima a la distribución real.

Para profundizar el análisis, la [Figura 6.8](#) presenta las matrices de confusión normalizadas por categoría racial correspondientes a DeepFace y al modelo FairFace modificado. En la matriz de DeepFace se evidenció un rendimiento desigual entre categorías, con fuertes concentraciones de errores en las razas india y otras (latina y medio oriental). En ambos casos, la proporción de aciertos desciende por debajo de un tercio. Asimismo, se evidenció una marcada tendencia del modelo a confundir la categoría otras con la clase blanca, lo que evidencia un patrón claro de sesgo.

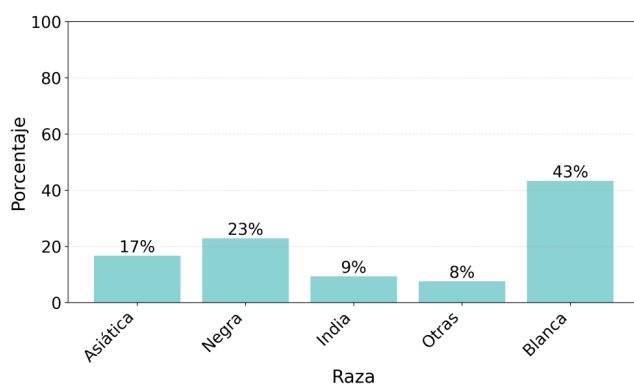
En contraste, la matriz correspondiente al modelo FairFace modificado exhibió un desempeño más equilibrado entre las distintas razas. Aunque persistieron las dificultades en categorías menos representadas, la proporción de aciertos fue sustancialmente mayor en cuatro de las cinco categorías evaluadas. Además, se redujo la dispersión de errores hacia categorías no relacionadas, lo que indica una mitigación de los sesgos reportados previamente en DeepFace. La estabilidad en el comportamiento del clasificador FairFace modificado, junto con su notable reducción en los tiempos de ejecución, sustenta la elección del modelo modificado para su incorporación en el pipeline.

6.2.3. Cálculo experimental del umbral de FairFace

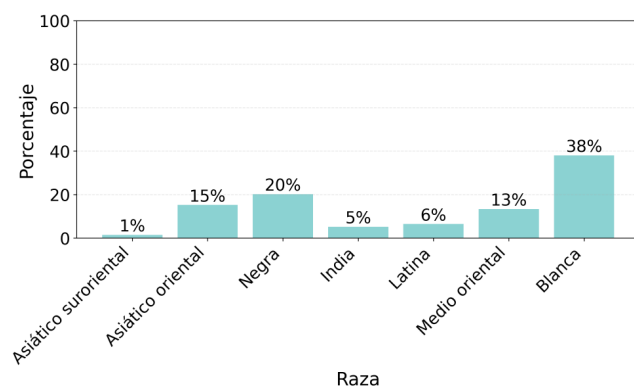
Las distancias obtenidas para cada predicción fueron analizadas mediante pruebas de normalidad con el fin de determinar si seguían una distribución normal. Se aplicó el test de normalidad de Kolmogorov-Smirnov, el cual rechazó la hipótesis de normalidad. Por lo tanto, fue necesario emplear criterios no paramétricos para la selección del umbral.

Los valores correspondientes a los cuartiles y al rango intercuartílico se presentan en la [Tabla 6.3](#). A partir de los resultados, y considerando la asimetría de la distribución, se seleccionó el tercer cuartil ($Q3 = 0,82$) como candidato para definir el umbral de aceptación. Para evaluar su efectividad, se utilizaron las imágenes del conjunto de validación de FairFace junto con sus etiquetas de raza originales y etiquetas aleatorias y se compararon con la predicción generada por el modelo. Este procedimiento permitió cuantificar el comportamiento del umbral propuesto y calcular el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Los valores obtenidos se resumen en la [Figura 6.9](#).

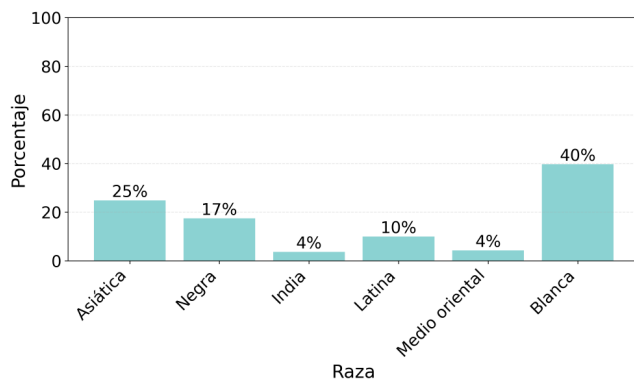
A partir de la matriz de confusión correspondiente al umbral seleccionado, se calcularon las métricas de precisión, recall, exactitud y F1-score, presentadas en la [Tabla 6.3](#). Los resultados confirmaron que el uso del tercer cuartil como umbral proporcionó un rendimiento adecuado con una separación efectiva entre predicciones correctas e incorrectas y resultó satisfactorio para la clasificación de raza en este contexto experimental.



(a) Distribución original del conjunto UTKFace.



(b) Distribución estimada por FairFace modificado.



(c) Distribución estimada por DeepFace.

Figura 6.7: Comparación de distribuciones raciales entre UTKFace, FairFace modificado y DeepFace.

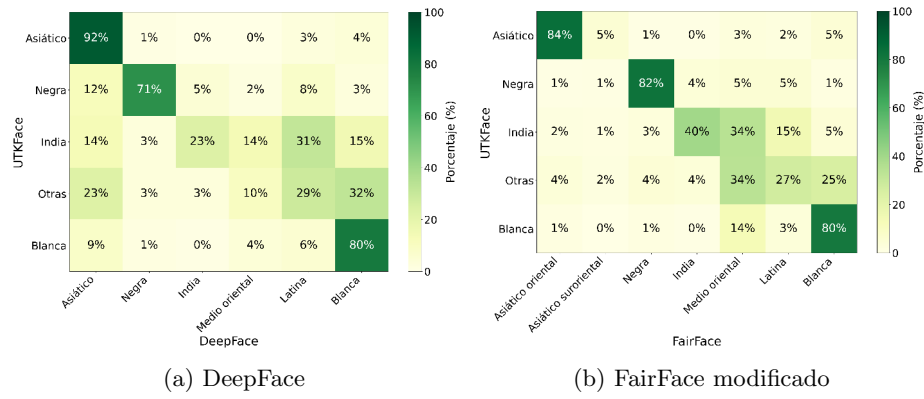


Figura 6.8: Matrices de confusión por raza en UTKFace generadas por DeepFace y por el modelo FairFace modificado.

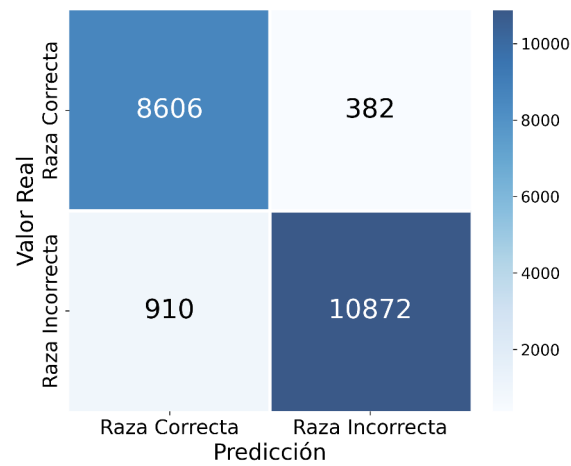


Figura 6.9: Matriz de confusión obtenida aplicando el umbral seleccionado sobre las predicciones del clasificador de raza.

<i>Métricas de desempeño</i>		<i>Estadísticos de distribución</i>	
<i>Métrica</i>	<i>Valor</i>	<i>Métrica</i>	<i>Valor</i>
Precision	90.44 %	Q1	0.0600
Recall	95.75 %	Q2	0.3374
Accuracy	93.78 %	Q3	0.8196
F1-score	93.02 %	IQR	0.7596

Tabla 6.3: Resultados obtenidas al aplicar el umbral seleccionado sobre las distancias del clasificador de raza.

6.2.4. Evaluación del sesgo en la generación de rostros con StyleGAN3

La [Figura 6.10](#) presenta la distribución de razas estimada por FairFace a partir de 150.000 rostros generados por StyleGAN3. Los resultados mostraron un predominio de individuos clasificados como rostros de raza blanca, que representaron un 75,2 % del total. La segunda raza con mayor presencia correspondió a la latina con un 18,7 %. La raza que siguió es la asiático oriental con una representación del 4 %. El resto de las razas resultaron en porcentajes considerablemente más bajos e inferiores al uno por ciento.

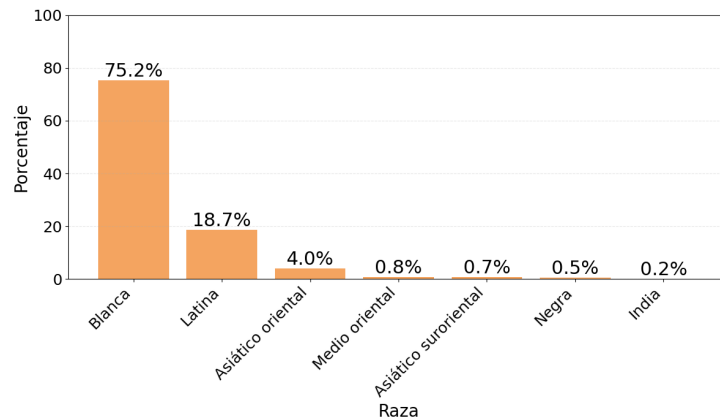


Figura 6.10: Distribución de raza según StyleGAN3.

En función de los resultados de la evaluación de FairFace, la distribución de la [Figura 6.10](#) no pudo atribuirse exclusivamente a errores de clasificación de FairFace. Aunque FairFace presentó un pequeño error de clasificación en categorías poco representadas, su comportamiento en datos reales no produjo una asimetría tan marcada como la que aparece en los rostros generados. Por lo tanto, la fuerte concentración en la raza blanca y la subrepresentación del resto de las razas indicaron la existencia de un desbalance proveniente principalmente de la generación de rostros con StyleGAN3.

6.3. Evaluación del algoritmo multiobjetivo

Esta sección presenta los resultados obtenidos en la evaluación del MOEA. La evaluación se desarrolló en tres etapas. En primer lugar, se llevaron a cabo experimentos preliminares destinados a estabilizar los parámetros generales del algoritmo. Posteriormente, se compararon distintas configuraciones paramétricas con el fin de calibrar las probabilidades de aplicabilidad de los operadores evolutivos. Finalmente, se realizó la validación del algoritmo para analizar su convergencia y la calidad de las soluciones generadas.

6.3.1. Experimentación preliminar

Esta sección presenta los resultados de los experimentos preliminares realizados sobre el tamaño de población y la cantidad de generaciones.

Experimentos sobre el tamaño de la población

La [Tabla 6.4](#) presenta los valores de hipervolumen relativo obtenidos para las configuraciones de 56 y 80 individuos para las instancias I1 y I3. Para cada configuración, el test de normalidad no rechazó la hipótesis nula, por lo que los resultados se reportaron mediante la media y la desviación estándar. Los resultados mostraron que la configuración con 80 individuos alcanzó valores de hipervolumen relativo superiores en promedio, además de una variabilidad menor entre ejecuciones.

<i>Instancia</i>	<i>Configuración</i>	<i>Media</i>	<i>Std</i>	<i>Min</i>	<i>Max</i>
I1	Población de 56	0.8654	0.0504	0.7566	0.9443
	Población de 80	0.9129	0.0370	0.8426	0.9995
I3	Población de 56	0.9227	0.0236	0.8663	0.9608
	Población de 80	0.9500	0.0181	0.9206	0.9994

Tabla 6.4: Resultados del hipervolumen relativo para las instancias I1 e I3 según el tamaño de la población.

Experimentos sobre la cantidad de generaciones

El objetivo del análisis consistió en determinar una cantidad de generaciones suficiente para asegurar la convergencia del algoritmo sin comprometer el tiempo computacional. Para las instancias I1, I2, I3 e I4 se realizaron treinta ejecuciones independientes utilizando la configuración presentada en la [Tabla 6.5](#).

<i>Parámetro</i>	<i>Valor</i>
Cantidad máxima de generaciones	200
Tamaño de población	80
Probabilidad de cruzamiento	0.9
Probabilidad de mutación	0.01
Blend-Alpha	0.2

Tabla 6.5: Configuración paramétrica para comparación de bibliotecas.

Con el fin de contar con una referencia consistente para cada instancia, se construyó un frente de Pareto combinado a partir de las soluciones no dominadas obtenidas en las treinta ejecuciones. El hipervolumen del frente combinado se empleó como valor de referencia para calcular la evolución del hipervolumen relativo a lo largo de las generaciones. La [Figura 6.11](#) presenta un incremento pronunciado del hipervolumen relativo en las primeras generaciones de la instancia I2, seguido de una estabilización progresiva a medida que el algoritmo se aproxima a la convergencia. El Anexo [A.1](#) presenta la evolución del hipervolumen relativo para otras instancias.

El comportamiento en la generación 100 fue coherente en todas las instancias evaluadas ya que la mejora porcentual del hipervolumen relativo se mantuvo por debajo del 0,6 %, lo que evidenció que el avance del algoritmo a partir de ese punto fue marginal y que continuar la evolución más allá las 100 generaciones no produjo mejoras significativas. La tabla [Tabla 6.6](#) presenta la mejora porcentual del hipervolumen relativo para la instancia I2.

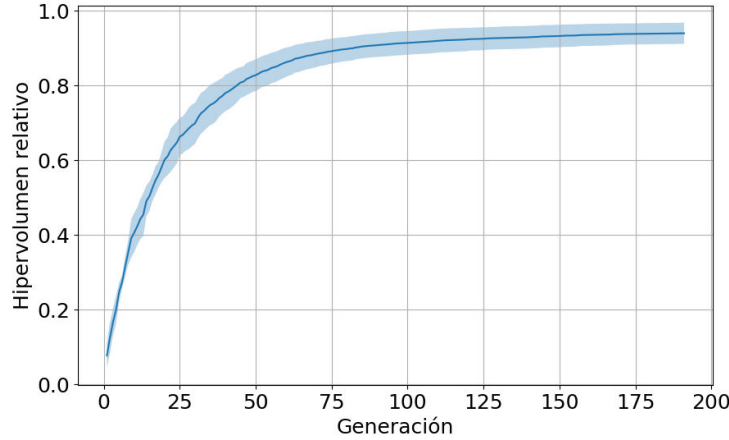


Figura 6.11: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I2.

<i>Intervalo de generaciones</i>	<i>Mejora porcentual</i>
1 – 10	430.52 %
11 – 20	42.21 %
21 – 30	14.45 %
31 – 40	9.27 %
41 – 50	5.59 %
51 – 60	3.66 %
61 – 70	2.25 %
71 – 80	1.30 %
81 – 90	1.00 %
91 – 100	0.61 %
101 – 110	0.56 %
111 – 120	0.36 %
121 – 130	0.31 %
131 – 140	0.21 %
141 – 150	0.30 %
151 – 160	0.27 %
161 – 170	0.20 %
171 – 180	0.11 %
181 – 190	0.10 %

Tabla 6.6: Mejora porcentual del hipervolumen relativo cada diez generaciones para la instancia I2.

6.3.2. Configuración paramétrica

Esta sección presenta los resultados obtenidos en los experimentos destinados a determinar los valores óptimos de la probabilidad de mutación y la probabilidad de cruzamiento del algoritmo evolutivo.

Las nueve configuraciones definidas en la [Tabla 5.3](#) fueron evaluadas sobre las instancias I1, I2, I3 e I4, realizando treinta ejecuciones independientes por cada configuración. Para cada instancia, se calculó el frente de Pareto de referencia a partir de las soluciones no dominadas de todas las ejecuciones y todas las configuraciones. Luego, se obtuvo el punto de referencia y el hipervolumen de referencia que fueron utilizados para calcular el hipervolumen relativo. Los valores completos correspondientes a cada ejecución se encuentran detallados en el Anexo [A.2](#).

Para analizar el comportamiento de las distribuciones de hipervolumen relativo, se aplicó el test de normalidad de Kolmogorov–Smirnov siguiendo el procedimiento descrito en la [Subsección 5.5.2](#). Los resultados obtenidos indicaron niveles de significancia superiores al umbral establecido en tres de las instancias evaluadas, por lo que no se rechazó la hipótesis de normalidad en esos casos.

En contraste, para la instancia I4 el test rechazó la hipótesis de normalidad. No obstante, con el propósito de asegurar la consistencia del análisis entre las cuatro instancias, se adoptaron la media y la desviación estándar como métricas representativas para evaluar el desempeño de las configuraciones analizadas. Las Tablas 6.8 y 6.9 presentan un resumen de los resultados para cada una de las instancias consideradas.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>
<i>I1</i>	0.7443	0.8650	0.4525	0.4960	0.9393	0.9513	0.9352	0.9784	0.7403
<i>I2</i>	0.8882	0.7404	0.6499	0.6062	0.9978	0.2850	0.2089	0.8883	0.6764
<i>I3</i>	0.9651	0.4938	0.4042	0.8862	0.5703	0.8100	0.6389	0.9172	0.8771

Tabla 6.7: Resultados del p – *valor* del test de normalidad de Kolmogorov-Smirnov para las instancias I1, I2, I3.

Configuración	I1				I2			
	Máximo	Mínimo	Media	Std	Máximo	Mínimo	Media	Std
C1	0.9743	0.7074	0.8434	0.0535	0.8810	0.7740	0.8310	0.0299
C2	0.9144	0.6845	0.8229	0.0583	0.9080	0.7330	0.8260	0.0436
C3	0.9191	0.6521	0.8417	0.0574	0.9120	0.7090	0.8370	0.0464
C4	0.9431	0.7119	0.8680	0.0559	0.9275	0.7548	0.8495	0.0434
C5	0.9448	0.7836	0.8705	0.0409	0.9309	0.7751	0.8473	0.0376
C6	0.9595	0.7750	0.8748	0.0441	0.9530	0.6980	0.8668	0.0499
C7	0.9988	0.8138	0.9094	0.0536	0.9568	0.7426	0.8783	0.0425
C8	0.9792	0.8038	0.9038	0.0417	0.9957	0.7921	0.8793	0.0550
C9	0.9999	0.8263	0.9029	0.0475	0.9410	0.7580	0.8727	0.0449

Tabla 6.8: Media, desviación estándar, mínimo y máximo del hipervolumen relativo obtenido para I1 e I2, para cada configuración.

Configuración	I3				I4			
	Máximo	Mínimo	Media	Std	Máximo	Mínimo	Media	Std
C1	0.9493	0.7910	0.8664	0.0364	0.9250	0.1846	0.8083	0.1693
C2	0.9440	0.7786	0.8705	0.0412	0.9188	0.1829	0.8138	0.1722
C3	0.9243	0.7651	0.8588	0.0472	0.9356	0.1749	0.8134	0.2119
C4	0.9860	0.8395	0.9001	0.0379	0.9518	0.1870	0.8697	0.1326
C5	0.9500	0.8119	0.8934	0.0366	0.9769	0.1843	0.8570	0.1327
C6	0.9627	0.8430	0.9102	0.0335	0.9771	0.1842	0.8710	0.1347
C7	0.9977	0.8320	0.9156	0.0329	0.9815	0.1904	0.8883	0.1324
C8	0.9800	0.8737	0.9130	0.0246	0.9644	0.1987	0.8842	0.1316
C9	0.9872	0.8682	0.9237	0.0260	0.9717	0.1926	0.8386	0.2165

Tabla 6.9: Media, desviación estándar, mínimo y máximo del hipervolumen relativo obtenido para I3 e I4, para cada configuración.

Posteriormente, se aplicó el test de Friedman con el propósito de ordenar las configuraciones de parámetros según su desempeño. Se obtuvo el rango promedio asociado a cada combinación en el conjunto de instancias. Los resultados del test se presentan en la [Tabla 6.10](#) y muestran el rango resultante para cada configuración derivado de los hipervolumenes obtenidos en las treinta ejecuciones. La clasificación en rangos permitió identificar un conjunto reducido de configuraciones con los mejores valores.

La configuración C7 obtuvo el mejor rango promedio en I2 e I4 y empató en primer lugar con la configuración C8 en I1; en I3 la mejor configuración fue C9, aunque la configuración C7 quedó en segundo lugar y dentro del mismo grupo superior. En función de estos resultados, los parámetros seleccionados para la etapa de validación se presentan en la [Tabla 6.11](#).

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	<i>C9</i>
I1	6.60	6.83	6.27	5.07	5.20	4.77	3.33	3.33	3.60
I2	6.23	6.57	5.90	5.20	5.57	3.97	3.53	4.17	3.87
I3	7.00	6.53	7.00	4.63	5.10	3.77	3.67	3.97	3.33
I4	7.00	6.63	5.57	4.60	5.13	4.40	3.57	3.93	4.17

Tabla 6.10: Promedio de rangos para cada configuración paramétrica de cada instancia.

<i>Parámetro</i>	<i>Valor</i>
Cantidad de generaciones	100
Individuos en la población	80
Probabilidad de cruzamiento	0.9
Probabilidad de mutación	0.001

Tabla 6.11: Configuración de parámetros del algoritmo evolutivo.

6.3.3. Validación

Esta sección presenta los resultados obtenidos y analiza el desempeño del algoritmo en las nueve instancias de validación, a partir de treinta ejecuciones independientes realizadas para cada una de las instancias.

Análisis de la convergencia del algoritmo

El hipervolumen relativo presentado en la [Tabla 6.12](#) permitió evaluar la convergencia del frente de Pareto obtenido en las treinta ejecuciones para cada instancia. Si bien el valor de hipervolumen de referencia es aproximado, los resultados de hipervolumen relativo evidenciaron un desempeño elevado del algoritmo a lo largo de todas las instancias evaluadas.

El spacing mostrado en la [Tabla 6.13](#) presenta valores bajos y muy similares entre todas las instancias, lo que indicó que la distribución de las soluciones de cada frente de Pareto fue uniforme.

El tiempo medio de ejecución por instancia, presentado en la [Tabla 6.14](#), se mantuvo en valores comparables a los reportados en trabajos previos. Este resultado evidenció que la incorporación de un segundo sistema de reconocimiento facial ejecutado en un proceso en paralelo no introdujo un incremento significativo en los tiempos de ejecución del pipeline.

La [Figura 6.12](#) muestra la evolución de la media y la desviación estándar del hipervolumen relativo a lo largo de las generaciones para la instancia I6. La [Figura 6.13](#) representa la evolución del frente de Pareto a lo largo de distintas generaciones. En las iteraciones iniciales, las soluciones se encontraron ampliamente distribuidas y mostraron valores altos en al menos una de las funciones objetivo. A medida que avanza el algoritmo evolutivo, los puntos se desplazan hacia regiones más favorables, reduciendo simultáneamente las distancias de similitud y raza.

<i>Instancia</i>	<i>Máximo</i>	<i>Mínimo</i>	<i>Media</i>	<i>Std</i>
I5	0.9939	0.9045	0.9522	0.0224
I6	0.9972	0.6897	0.9198	0.0540
I7	0.9900	0.7029	0.8677	0.0704
I8	0.9770	0.7508	0.8763	0.0492
I9	0.9940	0.8016	0.8809	0.0455
I10	0.9960	0.8826	0.9379	0.0329
I11	0.9961	0.7152	0.8752	0.0496
I12	0.9902	0.8225	0.9277	0.0420
I13	0.9748	0.7713	0.8878	0.0594

Tabla 6.12: Máximo, mínimo, media y desviación estándar del hipervolumen relativo para cada instancia.

<i>Instancia</i>	<i>Media</i>	<i>Std</i>
I5	0.0121	0.0028
I6	0.0145	0.0076
I7	0.0150	0.0065
I8	0.0141	0.0091
I9	0.0138	0.0049
I10	0.0144	0.0070
I11	0.0117	0.0120
I12	0.0160	0.0058
I13	0.0154	0.0075

Tabla 6.13: Media y desviación estándar del spacing por instancia.

<i>Instancia</i>	<i>Media (s)</i>	<i>Std (s)</i>	<i>Media (m)</i>	<i>Std (m)</i>
I5	11603	1195	193	20
I6	10596	2327	176	38
I7	13038	1040	217	17
I8	11551	1390	192	23
I9	13064	960	218	16
I10	13765	3209	229	54
I11	11943	1425	199	24
I12	11568	1402	192	23
I13	13315	1036	222	17

Tabla 6.14: Media y desviación estándar del tiempo de ejecución por instancia.

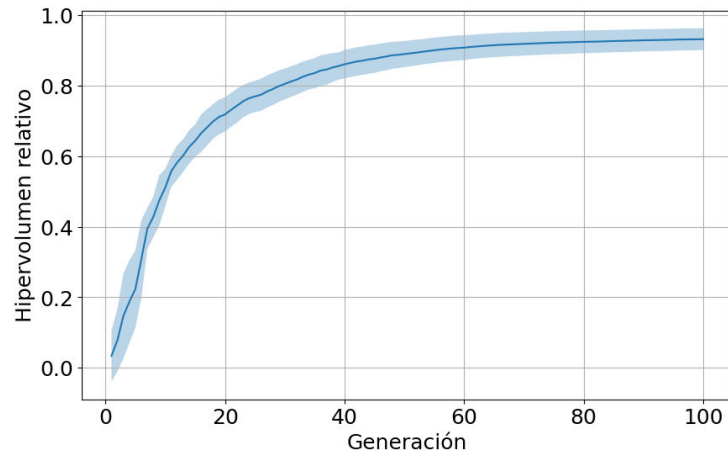


Figura 6.12: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I6.

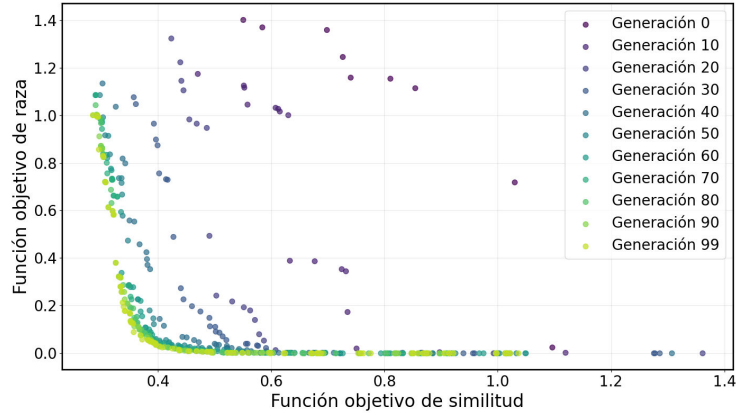


Figura 6.13: Evolución del frente de Pareto para la instancia I6.

Desempeño de los reconocedores faciales y de raza

Se evaluó la precisión de las soluciones no dominadas obtenidas en la etapa de validación mediante las tasas de acierto reportadas en la [Tabla 6.15](#). Se compararon los valores de distancia de identidad obtenidos por los reconocedores VGG-Face y Dlib, así como los valores de distancia racial calculados por FairFace para todas las soluciones no dominadas de las treinta ejecuciones. Se definieron como aciertos aquellas imágenes que alcanzaron una distancia inferior al umbral establecido para cada reconocedor en la [Subsección 6.1.2](#) y en el caso de la raza, aquellas que se ubicaron por debajo del umbral definido en la [Subsección 6.2.3](#). La métrica $A_{VGG-Face}$ refleja los aciertos obtenidos por VGG-Face, A_{Dlib} refleja los aciertos obtenidos por Dlib, A_{Rec} corresponde a los aciertos simultáneos en ambos reconocedores y A_{Raza} representa la tasa de imágenes que alcanzaron una distancia racial menor al umbral establecido.

Los valores obtenidos mostraron que el algoritmo conservó un nivel alto de acierto en la clasificación de raza para todas las instancias, a su vez los reconocedores faciales presentaron variabilidad según la combinación entre la raza inicial y la raza objetivo.

En las instancias I5, I10, I11 e I12 ambas métricas de similitud alcanzaron valores moderados, a su vez la tasa de acierto en raza fue elevada. Esto indicó que la transición hacia la raza objetivo se logró de forma consistente, aunque la diferencia entre la raza inicial y la raza objetivo pudo introducir variabilidad en el reconocimiento facial. Las instancias I6 e I8 presentaron altos niveles de acierto en ambos reconocedores, VGG-Face y Dlib, acompañados de una alta precisión en la clasificación racial, lo que sugirió que en estas configuraciones la preservación de similitud resultó más estable. Por su parte, las instancias I7 e I9 mostraron valores moderados en los aciertos de similitud y raza, lo que reflejó una transición más exigente para los reconocedores y una mayor sensibilidad del pipeline frente a estas combinaciones específicas de identidad inicial y raza objetivo.

<i>Instancia</i>	$A_{VGG-Face}$	A_{Dlib}	A_{Rec}	A_{Raza}
I5	72.58 %	77.17 %	71.83 %	87.54 %
I6	89.08 %	97.55 %	88.85 %	96.30 %
I7	81.41 %	77.34 %	71.58 %	79.37 %
I8	94.25 %	95.06 %	93.58 %	99.46 %
I9	69.90 %	78.20 %	67.98 %	77.04 %
I10	44.89 %	70.63 %	44.89 %	94.85 %
I11	63.25 %	78.26 %	62.94 %	98.15 %
I12	87.16 %	66.55 %	66.55 %	88.95 %
I13	35.93 %	63.37 %	35.89 %	85.41 %

Tabla 6.15: Aciertos de VGG-Face, Dlib y FairFace para cada instancia.

La [Figura 6.14](#) muestra la representación tridimensional de las soluciones no dominadas para la instancia I6 en el espacio definido por las distancias de similitud de VGG-Face y Dlib, junto con las distancias de cumplimiento de raza. La figura incluye también los planos correspondientes a los umbrales de cada métrica, lo que permite visualizar de manera conjunta las tres regiones factibles del problema.

Para las instancias I6 e I8, las [Figuras 6.15](#) y [6.16](#) ilustran las soluciones no dominadas obtenidas en las treinta ejecuciones. En cada figura se destacan los cuadrantes determinados por los umbrales definidos. La región marcada en rojo corresponde a la zona con valores menores al umbral de raza y las zonas en naranja y violeta corresponden a los valores menores a los umbrales de VGG-Face y Dlib respectivamente.

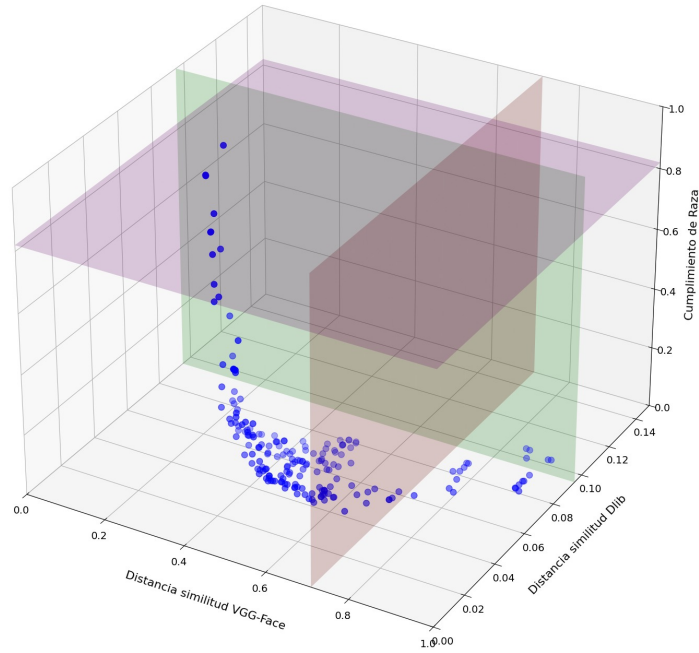
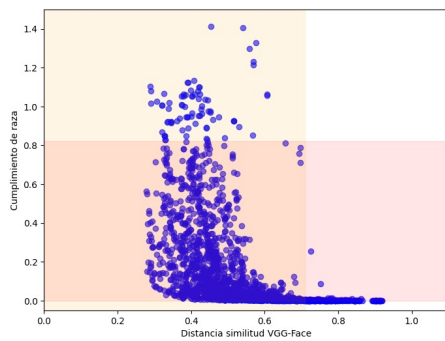
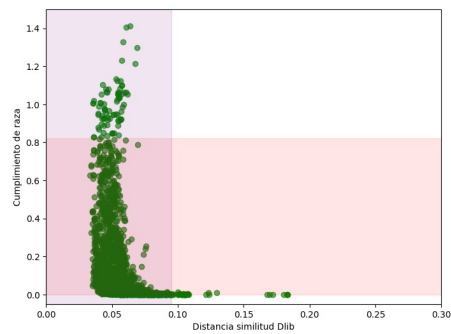


Figura 6.14: Soluciones no dominadas con respecto a los ejes de similitud VGG-Face, similitud Dlib y raza para la instancia I6.



(a) Similitud VGG-Face vs. raza



(b) Similitud Dlib vs. raza

Figura 6.15: Soluciones no dominadas para la instancia I6.

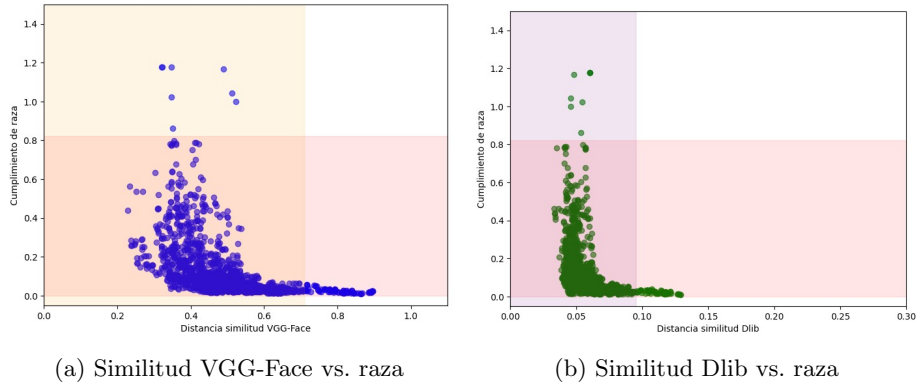


Figura 6.16: Soluciones no dominadas para la instancia I8.

Rostros generados por el algoritmo

La [Figura 6.17](#) muestra cinco imágenes representativas del frente de Pareto de referencia para la instancia I7 ordenadas de menor a mayor por distancia de similitud, de modo que al aumentar la distancia de similitud disminuye la distancia de cumplimiento de raza. La [Tabla 6.16](#) presenta las distancias de similitud obtenidas por VGG-Face y Dlib para cada imagen y la [Tabla 6.17](#) reporta la distribución racial resultante del clasificador FairFace para las cinco imágenes. Las imágenes generadas evidenciaron cómo el algoritmo mejoró uno de los objetivos mientras cedió en otro, reflejando el compromiso de las soluciones que conforman un frente de Pareto. El análisis para las otras instancias se encuentra en el Anexo [B.4](#).



Figura 6.17: Instancia I12 e imágenes representativas del frente de Pareto de referencia.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3217	0.4800	0.4617	0.4427	0.5476
Dlib	0.0405	0.0372	0.0369	0.0466	0.0778

Tabla 6.16: Distancias de similitud de las cinco imágenes representativas de la instancia I12.

<i>Raza</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0	0.37	0.82	0.99	1.00
Negra	0.71	0.14	0.01	0	0
Latina	0.02	0.33	0.13	0.01	0
Asiático oriental	0	0	0	0	0
Asiático suroriental	0	0.00	0.00	0	0
India	0.27	0.11	0.01	0	0
Medio oriental	0.00	0.05	0.03	0	0

Tabla 6.17: Distribución racial de las cinco imágenes representativas de la instancia I12.

Calidad de resultados

Para evaluar la calidad de las soluciones obtenidas, se seleccionaron cinco imágenes alternativas para cada instancia y se compararon con las cinco soluciones representativas del frente de Pareto de referencia presentadas en la [Sección 6.3.3](#). Adicionalmente, se incorporaron cinco soluciones seleccionadas aleatoriamente del mismo frente, con el fin de ampliar la cobertura del análisis. Para cada uno de estos conjuntos, se calcularon las distancias de similitud con los reconocedores VGG-Face y Dlib y los resultados se visualizaron en forma de heatmaps independientes por sistema de reconocimiento. La [Figura 6.20](#) muestra los resultados para la instancia I11. El resto de las instancias se presentan en el [Anexo B.4](#). En todos los casos, los patrones mostraron que las soluciones mantuvieron distancias uniformes dentro de cada reconocedor.

De forma complementaria, se realizó un análisis con reconocedores de estado del arte que no formaron parte del pipeline. El objetivo del análisis fue evaluar la capacidad de generalizar las soluciones obtenidas. Se consideraron todas las soluciones del frente de Pareto de referencia y se seleccionó el subconjunto de imágenes que lograron engañar al menos a uno de los reconocedores del pipeline. El subconjunto se evaluó con cuatro modelos adicionales de reconocimiento facial: Facebook DeepFace, Facenet, ArcFace y SFace. En todos los casos, se utilizó el modelo de detección OpenCV. Los resultados, presentados en la [Tabla 6.18](#) mostraron un desempeño elevado para varios de estos reconocedores, con tasas de acierto cercanas o superiores al 80 % en múltiples instancias. Aunque algunos modelos presentaron comportamientos más conservadores, el desempeño general sugirió que las soluciones generadas no dependieron exclusivamente de las características particulares de VGG-Face o Dlib, sino que mantuvieron rasgos suficientemente robustos para transferirse a arquitecturas diferentes.



Figura 6.18: Instancia I11 e imágenes representativas del frente de Pareto de referencia.

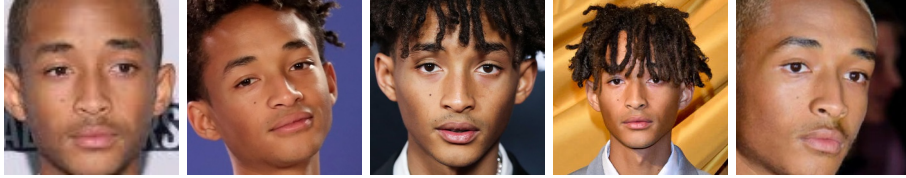


Figura 6.19: Imágenes alternativas a la instancia I11.

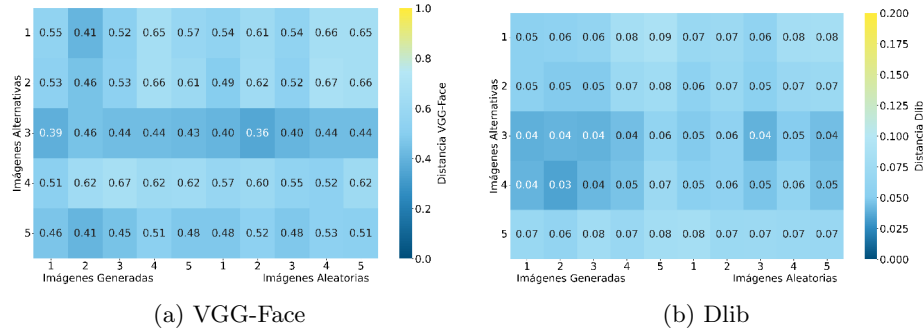


Figura 6.20: Heatmaps de distancias de similitud para la instancia I11.

<i>Instancia</i>	<i>DeepFace</i>	<i>FaceNet</i>	<i>ArcFace</i>	<i>SFace</i>
I5	92.65 %	45.59 %	100 %	36.76 %
I6	78.12 %	78.12 %	84.38 %	95.31 %
I7	88.10 %	23.81 %	60.71 %	46.43 %
I8	98.11 %	35.85 %	88.68 %	88.68 %
I9	0 %	4.71 %	12.94 %	16.47 %
I10	25 %	4.41 %	85.29 %	20.59 %
I11	44 %	4 %	100 %	94 %
I12	86.15 %	24.62 %	89.23 %	60 %
I13	98.75 %	11.25 %	67.5 %	0 %

Tabla 6.18: Porcentaje de aciertos por modelo de reconocimiento facial.

Capítulo 7

Conclusiones y trabajo futuro

Este capítulo presenta las conclusiones generales del proyecto de grado. Asimismo, se discuten posibles líneas de investigación futura que podrían extender, profundizar o complementar la solución presentada.

7.1. Conclusiones

El proyecto abordó el estudio, diseño e implementación de un sistema basado en un MOEA orientado a la exploración del espacio latente de GANs. El propósito central fue generar imágenes sintéticas que conservaran la mayor similitud posible con una imagen objetivo, garantizando a la vez el cumplimiento con características de una raza predefinida. El trabajo se desarrolló como una extensión de proyectos previos que lograron guiar la búsqueda del espacio latente de una GAN optimizando a la vez atributos de raza, género y edad utilizando un reconocedor facial y un clasificador de atributos. Este proyecto propuso la incorporación de un segundo reconocedor, con el fin de reducir el acoplamiento de las soluciones a un único sistema de reconocimiento y fortalecer la robustez del pipeline frente a sesgos y variaciones propias de cada arquitectura de reconocimiento facial. Además, se integró una biblioteca alternativa para la extracción de características raciales con el propósito de mitigar el sesgo racial identificado en proyectos previos.

La investigación de trabajos relacionados permitió establecer un marco conceptual para el desarrollo del proyecto. El análisis incluyó investigaciones sobre algoritmos evolutivos aplicados a la exploración del espacio latente de GANs para la generación de rostros sintéticos, así como estudios orientados a caracterizar los sesgos presentes en clasificadores faciales y a comparar distintas configuraciones de sistemas de reconocimiento. En conjunto, la literatura revisada validó la pertinencia del enfoque evolutivo y aportó fundamentos empíricos para la selección de VGG-Face y Dlib como modelos de reconocimiento, además de jus-

tificar la incorporación de FairFace como clasificador racial entrenado con un conjunto de datos menos sesgado.

El pipeline desarrollado en este proyecto se estructuró a partir de cuatro componentes principales: StyleGAN3 para la generación de rostros sintéticos, la biblioteca DeepFace, que encapsula los dos sistemas de reconocimiento, FairFace, que implementa el modelo de clasificación racial, y la biblioteca DEAP para la implementación del algoritmo evolutivo. El proceso comienza con la generación de una imagen mediante StyleGAN3, que posteriormente es procesada de manera independiente por los dos sistemas de reconocimiento facial. Cada uno detecta, alinea y recorta el rostro utilizando sus propios modelos de preprocesamiento para luego extraer los embeddings con el reconocedor correspondiente. El primer sistema emplea MtCnn como detector y VGG-Face como modelo de reconocimiento. El segundo utiliza los modelos especializados provistos por la biblioteca Dlib para la detección y para el reconocimiento. Para obtener el vector de características de raza se utiliza el modelo de clasificación de siete razas de FairFace. Las medidas de distancia obtenidas al comparar los embeddings generados con los embeddings objetivo se utilizan para evaluar el fitness de cada individuo dentro del algoritmo evolutivo multiobjetivo, que actualiza la población aplicando operadores de cruce y mutación a lo largo de múltiples generaciones.

Para el análisis del sesgo del generador y el reconocedor facial, se utilizó StyleGAN3 para generar 150 mil imágenes y se evaluaron con FairFace. El estudio de la distribución resultante evidenció desequilibrios en la frecuencia de ciertas categorías raciales, donde las razas blanca y latina fueron las más favorecidas. Estos hallazgos resultaron fundamentales para contextualizar los resultados del algoritmo y para comprender las limitaciones asociadas con la generación de rostros sintéticos con atributos raciales predeterminados.

El análisis realizado para determinar los umbrales de reconocimiento permitió establecer un criterio cuantitativo para distinguir pares de imágenes pertenecientes a la misma persona de personas diferentes. Se realizaron comparaciones utilizando imágenes del conjunto LFW y se evaluaron las distancias obtenidas mediante un árbol de decisión y KDE. En conjunto, estos procedimientos permitieron identificar los umbrales de reconocimiento facial empleados en los experimentos posteriores. Para poder evaluar dos reconocedores dentro de una misma función objetivo, fue necesario normalizar las distancias generadas por los dos sistemas de reconocimiento facial. Se definió un factor de normalización que permitió alinear ambas escalas de reconocimiento, garantizando una contribución equilibrada de cada reconocedor dentro del proceso evolutivo.

Para la selección del clasificador de raza se realizaron experimentos para evaluar el sesgo presente en FairFace comparando su desempeño con el clasificador provisto por la biblioteca DeepFace. Se empleó el conjunto UTKFace, que cuenta con imágenes etiquetadas con sus categorías raciales, lo que permitió analizar las predicciones de ambos clasificadores y evaluarlas con la etiqueta real. Los resultados mostraron que FairFace ofreció un mejor rendimiento, especialmente en la identificación de razas menos representadas. Esta diferencia motivó su incorporación como clasificador racial dentro del pipeline del proyecto.

Finalmente, para evaluar el EA, se realizó la etapa de ajuste de parámetros, en la cual se exploraron distintas configuraciones del algoritmo para identificar los valores que ofrecían la mejor calidad de soluciones. Las evaluaciones preliminares realizadas sobre los tamaños de población permitieron identificar el impacto de este parámetro en la calidad y en el costo computacional del algoritmo. Los resultados mostraron que el incremento en el número de individuos por generación justificaba el aumento en el tiempo de ejecución. Del mismo modo, el análisis experimental destinado a determinar la cantidad adecuada de generaciones permitió establecer un límite razonable para las ejecuciones del algoritmo evolutivo. Para los análisis siguientes, se seleccionaron 13 imágenes de rostros considerando la diversidad en género, raza y edad. Cuatro imágenes fueron utilizadas en la etapa de configuración de parámetros y las nueve restantes se reservaron para el proceso de validación.

La configuración paramétrica permitió identificar la combinación de probabilidades de cruzamiento y mutación que ofreció el mejor desempeño del algoritmo evolutivo. Para ello, se evaluaron nueve configuraciones posibles sobre las cuatro instancias seleccionadas utilizando la métrica de hipervolumen. Los resultados fueron analizados mediante el test de normalidad de Kolmogorov-Smirnov y el test de rangos de Friedman. A partir de este análisis, fue posible seleccionar la combinación más adecuada de probabilidades de cruzamiento y mutación.

La evaluación final del pipeline permitió validar el desempeño del algoritmo. Las treinta ejecuciones independientes realizadas para cada instancia mostraron una convergencia estable, respaldada por los valores del hipervolumen relativo. La inclusión del spacing permitió complementar este análisis al mostrar que las soluciones no dominadas mantenían una distribución adecuada a lo largo del frente. Asimismo, el estudio del progreso por generación confirmó que el algoritmo construye gradualmente frentes más cercanos al frente de referencia. En cuanto al comportamiento de los reconocedores integrados en el pipeline, las tasas de acierto obtenidas para VGG-Face, Dlib y FairFace demostraron que las soluciones generadas cumplen con los objetivos de similitud y raza. Además, el análisis realizado utilizando fotografías alternativas de los individuos objetivo mostró un comportamiento consistente con los resultados obtenidos a partir de la imagen objetivo original. Finalmente, la experimentación con reconocedores de estado del arte externos al pipeline evidenció la capacidad del algoritmo para generar imágenes capaces de engañar a múltiples sistemas de reconocimiento facial, lo que refuerza la generalización del ataque y confirma la efectividad del enfoque evolutivo desarrollado.

En resumen, la incorporación de un segundo reconocedor al algoritmo evolutivo permitió reducir el acoplamiento del sistema a un único sistema de reconocimiento facial. Las evaluaciones finales demostraron que el algoritmo converge de manera estable, produce frentes de Pareto diversos y mantiene un comportamiento robusto frente a variaciones en la instancia objetivo. Asimismo, utilizando las soluciones generadas, se logró engañar a cuatro reconocedores de estado del arte, lo que evidencia la generalización del ataque.

7.2. Trabajo futuro

El trabajo desarrollado abre diversas líneas de investigación futura orientadas a la optimización del algoritmo evolutivo y a la reducción del sesgo presente en los modelos de generación, clasificación y reconocimiento facial. Este trabajo consideró la evaluación combinada de VGG-Face y Dlib como reconocedores. Como trabajo futuro, se propone incorporar un tercer sistema de reconocimiento facial al pipeline. Esta integración puede abordarse de distintas maneras, una posibilidad consiste en incorporar la nueva métrica directamente dentro de la función objetivo de similitud facial y otra posibilidad es tratar cada sistema de reconocimiento como un objetivo independiente dentro de un algoritmo evolutivo estrictamente multiobjetivo.

Otra línea de trabajo relevante consiste en extender la optimización hacia nuevos atributos evaluados por FairFace. Si bien el presente estudio se centró en la categoría racial, es factible incorporar el género y la edad como objetivos adicionales. La inclusión simultánea de estos atributos demandaría un análisis más detallado del impacto del sesgo en las predicciones.

La eficiencia computacional también constituye un aspecto central a mejorar. El tiempo total de ejecución del algoritmo depende de manera significativa del modelo utilizado para la detección y para el reconocimiento. Explorar detectores alternativos más livianos puede ser una alternativa para reducir los cuellos de botella.

Otra línea de investigación relevante consiste en reconsiderar el modelo generativo empleado. En este trabajo se corroboró la presencia de un sesgo en StyleGAN3, reflejado en la generación predominante de rostros de personas de raza blanca. Se propone explorar arquitecturas generativas alternativas que presenten menor sesgo demográfico, o bien entrenar una variante de StyleGAN3 utilizando un conjunto de datos más equilibrado en términos de raza.

Finalmente, se propone explorar alternativas a la inicialización aleatoria de la población inicial y emplear una estrategia de inicialización inteligente, basada en la selección de individuos ubicados en direcciones del espacio latente más alineadas con los objetivos de búsqueda.

Referencias

- Bartz-Beielstein, T., Branke, J., Mehnen, J., y Mersmann, O. (2014). Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3), 178–195.
- Correa, J., Mignaco, J., Rey, G., Machín, B., Nesmachnow, S., y Toutouh, J. (2023). Multiobjective evolutionary search of the latent space of generative adversarial networks for human face generation. En *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (pp. 1768–1776). Association for Computing Machinery.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.
- Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., y Zafeiriou, S. (2022). ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 5962–5979.
- Eiben, A., y Smith, J. (2015). *Introduction to evolutionary computing*. Berlin, Heidelberg: Springer.
- Eshelman, L., y Schaffer, J. (1993). Real-coded genetic algorithms and the BLX- α crossover. En *Foundations of Genetic Algorithms* (pp. 187–202). Morgan Kaufmann.
- Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems*.
- He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- Kärkkäinen, K., y Joo, J. (2021). FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. En *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1548–1558).
- Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., y Aila, T. (2021). Alias-free generative adversarial networks. En *Advances*

in *Neural Information Processing Systems*.

- Karras, T., Laine, S., y Aila, T. (2019). A style-based generator architecture for generative adversarial networks. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4401–4410).
- King, D. (2015). *Trained model files for dlib example programs*. <https://github.com/davisking/dlib-models>. (Accedido: 2025-11-23)
- LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.
- Machín, B., Nesmachnow, S., y Toutouh, J. (2021). Evolutionary latent space search for driving human portrait generation. En *Proceedings of the IEEE Latin American Conference on Computational Intelligence* (pp. 1–6).
- Machín, B., Nesmachnow, S., y Toutouh, J. (2022). Multi-target evolutionary latent space search of a generative adversarial network for human face generation. En *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1878–1886). Association for Computing Machinery.
- Mitchell, M. (1999). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
- Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.
- Mohsienuddin, S., y Sabri, M. (2020). Facial recognition technology. *International Journal of Innovations in Engineering Research and Technology*, 7(6), 176–183.
- Nesmachnow, S., y Iturriaga, S. (2019). Cluster-UY: Collaborative scientific high performance computing in uruguay. En M. Torres y J. Klapp (Eds.), *Supercomputing* (pp. 188–202). Cham: Springer International Publishing.
- Nesmachnow, S., y Moreno, P. (2025). Three-objective evolutionary search of the latent space of generative adversarial networks for human face generation. En *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 2037–2045). Association for Computing Machinery.
- Parkhi, O., Vedaldi, A., y Zisserman, A. (2015). *VGG Face descriptor*. https://www.robots.ox.ac.uk/~vgg/software/vgg_face/. (Visual Geometry Group, University of Oxford. Accedido: 2025-11-22)
- Schroff, F., Kalenichenko, D., y Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815–823).
- Serengil, S., y Ozpinar, A. (2020). LightFace: A hybrid deep face recognition framework. En *Proceedings of the Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1–5).
- Serengil, S., y Ozpinar, A. (2024). A benchmark of facial recognition pipelines and co-usability performances of modules. *Bilisim Teknolojileri Dergisi*, 17(2), 95–107.
- Taigman, Y., Yang, M., Ranzato, M., y Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1701–1708).

- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4, 65–85.
- Zhang, K., Zhang, Z., Li, Z., y Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
- Zhang, Z., Song, Y., y Qi, H. (2017). *UTKFace: A large-scale face dataset for age, gender, and ethnicity*. <https://github.com/jackzhenguo/utkface>. (Repositorio de GitHub. Accedido: 2025-11-24)
- Zhong, Y., Deng, W., Hu, J., Zhao, D., Li, X., y Wen, D. (2021). SFace: Sigmoid-constrained hypersphere loss for robust face recognition. *IEEE Transactions on Image Processing*, 30, 2587–2598.

Anexo A

Tablas de configuración paramétrica

A.1. Experimentación preliminar

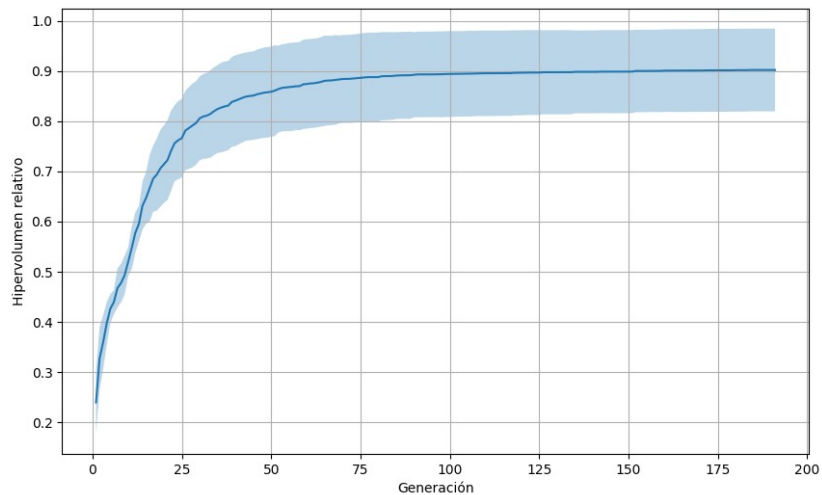


Figura A.1: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I1

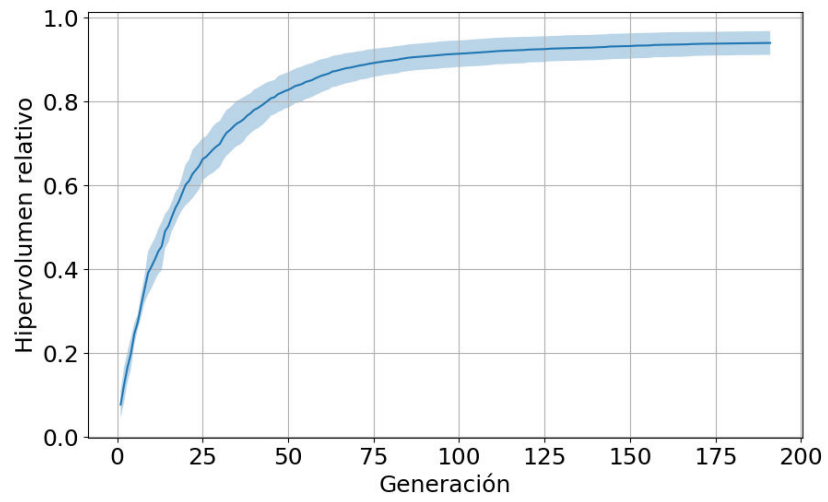


Figura A.2: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I2

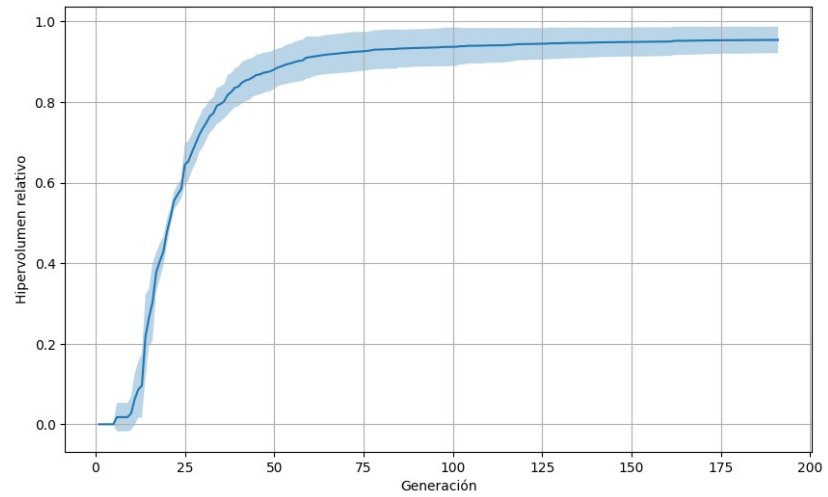


Figura A.3: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I3

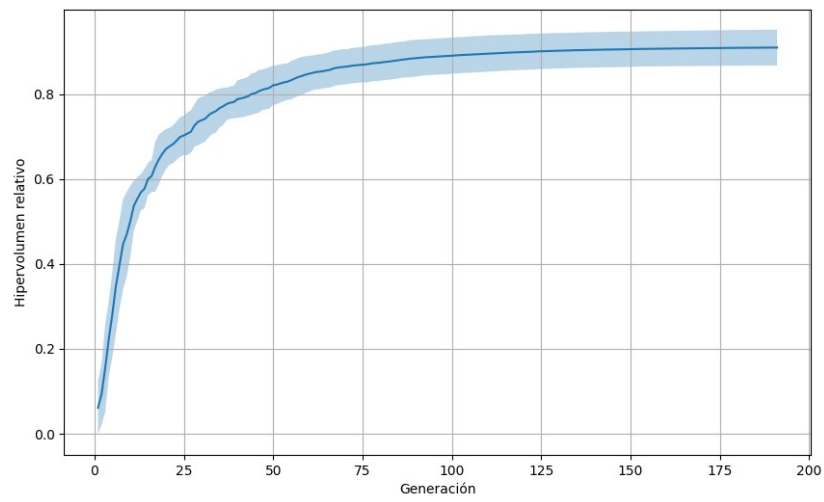


Figura A.4: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I4.

A.2. Resultados de configuración paramétrica

C1	C2	C3	C4	C5	C6	C7	C8	C9
0.825	0.745	0.875	0.748	0.894	0.924	0.870	0.902	0.898
0.802	0.877	0.834	0.870	0.854	0.909	0.861	0.890	0.844
0.829	0.809	0.862	0.802	0.801	0.896	0.994	0.916	0.826
0.871	0.848	0.826	0.899	0.884	0.874	0.963	0.944	0.838
0.801	0.914	0.898	0.931	0.811	0.884	0.940	0.904	0.840
0.879	0.889	0.868	0.833	0.878	0.960	0.900	0.871	0.934
0.870	0.685	0.849	0.924	0.832	0.913	0.906	0.804	0.900
0.870	0.851	0.875	0.913	0.936	0.895	0.910	0.979	0.919
0.872	0.905	0.840	0.908	0.885	0.882	0.852	0.898	0.852
0.954	0.750	0.913	0.867	0.850	0.794	0.983	0.971	0.901
0.824	0.893	0.864	0.850	0.919	0.894	0.951	0.950	0.873
0.883	0.855	0.729	0.712	0.945	0.810	0.946	0.910	0.876
0.816	0.779	0.918	0.908	0.784	0.888	0.910	0.920	0.949
0.839	0.778	0.845	0.859	0.842	0.842	0.850	0.924	0.878
0.785	0.817	0.866	0.903	0.892	0.959	0.814	0.890	0.892
0.848	0.879	0.906	0.817	0.796	0.818	0.907	0.911	0.927
0.764	0.768	0.886	0.874	0.884	0.877	0.922	0.812	0.999
0.873	0.734	0.809	0.925	0.909	0.855	0.961	0.934	0.974
0.974	0.789	0.832	0.809	0.886	0.924	0.978	0.961	0.890
0.853	0.745	0.830	0.819	0.890	0.898	0.874	0.855	0.904
0.853	0.792	0.784	0.926	0.884	0.885	0.942	0.876	0.904
0.879	0.807	0.828	0.877	0.864	0.818	0.832	0.878	0.929
0.891	0.874	0.652	0.943	0.870	0.859	0.990	0.951	0.917
0.833	0.838	0.919	0.902	0.931	0.775	0.999	0.908	0.868
0.886	0.849	0.783	0.898	0.858	0.927	0.907	0.944	0.991
0.803	0.831	0.891	0.894	0.824	0.874	0.819	0.854	0.832
0.772	0.862	0.772	0.853	0.873	0.862	0.886	0.868	0.955
0.707	0.842	0.878	0.839	0.848	0.853	0.846	0.918	0.876
0.784	0.777	0.825	0.937	0.925	0.869	0.849	0.865	0.995
0.860	0.909	0.793	0.798	0.871	0.829	0.923	0.908	0.904

Tabla A.1: Resultados de la configuración paramétrica para la instancia I1 con raza objetivo blanca.

C1	C2	C3	C4	C5	C6	C7	C8	C9
0.835	0.888	0.878	0.809	0.849	0.892	0.866	0.865	0.899
0.780	0.796	0.837	0.928	0.849	0.927	0.856	0.844	0.895
0.809	0.824	0.783	0.803	0.869	0.875	0.914	0.878	0.865
0.807	0.795	0.901	0.847	0.848	0.882	0.841	0.891	0.921
0.879	0.856	0.825	0.869	0.785	0.881	0.903	0.971	0.795
0.866	0.806	0.827	0.856	0.893	0.829	0.885	0.834	0.835
0.774	0.863	0.827	0.859	0.858	0.898	0.888	0.957	0.823
0.779	0.816	0.883	0.755	0.892	0.888	0.907	0.859	0.844
0.807	0.772	0.843	0.892	0.895	0.888	0.883	0.824	0.907
0.828	0.803	0.867	0.814	0.816	0.914	0.888	0.792	0.884
0.806	0.758	0.857	0.817	0.931	0.903	0.957	0.910	0.939
0.838	0.775	0.809	0.868	0.795	0.880	0.810	0.968	0.876
0.861	0.869	0.898	0.825	0.832	0.864	0.871	0.812	0.849
0.832	0.909	0.807	0.866	0.844	0.834	0.743	0.825	0.774
0.820	0.763	0.830	0.910	0.884	0.898	0.894	0.937	0.844
0.864	0.733	0.863	0.861	0.835	0.846	0.800	0.961	0.880
0.842	0.864	0.888	0.904	0.867	0.887	0.881	0.859	0.860
0.836	0.794	0.872	0.862	0.816	0.892	0.888	0.841	0.867
0.790	0.852	0.767	0.846	0.797	0.900	0.917	0.801	0.876
0.869	0.868	0.829	0.853	0.775	0.868	0.911	0.840	0.940
0.824	0.899	0.814	0.822	0.830	0.841	0.884	0.916	0.758
0.829	0.806	0.871	0.772	0.812	0.890	0.894	0.854	0.907
0.797	0.840	0.710	0.890	0.867	0.698	0.889	0.902	0.880
0.867	0.878	0.911	0.897	0.855	0.735	0.917	0.879	0.901
0.843	0.824	0.891	0.818	0.831	0.893	0.922	0.925	0.904
0.856	0.818	0.834	0.856	0.805	0.836	0.924	0.876	0.836
0.881	0.851	0.766	0.866	0.888	0.858	0.831	0.996	0.918
0.832	0.854	0.832	0.911	0.830	0.833	0.869	0.863	0.885
0.860	0.786	0.841	0.855	0.877	0.899	0.828	0.796	0.878
0.810	0.826	0.761	0.755	0.903	0.853	0.889	0.900	0.941

Tabla A.2: Resultados de la configuración paramétrica para la instancia I2 con raza objetivo asiática oriental.

C1	C2	C3	C4	C5	C6	C7	C8	C9
0.861	0.816	0.894	0.841	0.931	0.849	0.941	0.887	0.934
0.824	0.857	0.835	0.873	0.892	0.916	0.911	0.902	0.971
0.791	0.872	0.777	0.864	0.949	0.877	0.891	0.904	0.903
0.881	0.922	0.768	0.849	0.906	0.890	0.973	0.882	0.987
0.882	0.857	0.924	0.876	0.925	0.867	0.893	0.887	0.908
0.949	0.809	0.843	0.986	0.851	0.915	0.920	0.962	0.914
0.898	0.882	0.767	0.844	0.842	0.876	0.919	0.980	0.946
0.826	0.862	0.828	0.894	0.852	0.911	0.858	0.931	0.946
0.910	0.890	0.765	0.975	0.901	0.923	0.902	0.905	0.897
0.859	0.901	0.892	0.915	0.861	0.957	0.885	0.903	0.948
0.873	0.802	0.862	0.817	0.893	0.961	0.832	0.921	0.930
0.863	0.920	0.834	0.878	0.890	0.963	0.941	0.923	0.941
0.883	0.894	0.852	0.907	0.851	0.885	0.892	0.880	0.889
0.874	0.878	0.866	0.890	0.881	0.938	0.929	0.907	0.897
0.862	0.779	0.897	0.927	0.941	0.938	0.903	0.876	0.926
0.890	0.880	0.901	0.923	0.852	0.929	0.907	0.913	0.932
0.894	0.912	0.830	0.958	0.950	0.918	0.891	0.938	0.911
0.836	0.893	0.856	0.915	0.904	0.868	0.906	0.939	0.929
0.914	0.794	0.917	0.839	0.925	0.953	0.913	0.910	0.939
0.911	0.892	0.913	0.857	0.900	0.936	0.914	0.896	0.943
0.869	0.823	0.904	0.889	0.848	0.945	0.905	0.898	0.945
0.897	0.874	0.866	0.906	0.940	0.907	0.898	0.926	0.927
0.830	0.944	0.908	0.926	0.845	0.896	0.998	0.946	0.889
0.909	0.883	0.822	0.878	0.881	0.910	0.902	0.898	0.868
0.847	0.905	0.896	0.942	0.931	0.939	0.962	0.934	0.954
0.824	0.876	0.870	0.940	0.908	0.843	0.926	0.905	0.903
0.875	0.915	0.880	0.925	0.894	0.896	0.960	0.920	0.911
0.844	0.816	0.899	0.928	0.913	0.855	0.955	0.916	0.905
0.816	0.867	0.794	0.903	0.932	0.910	0.828	0.928	0.927
0.802	0.899	0.902	0.880	0.812	0.937	0.939	0.874	0.891

Tabla A.3: Resultados de la configuración paramétrica para la instancia I3 con raza objetivo india.

C1	C2	C3	C4	C5	C6	C7	C8	C9
0.859	0.859	0.837	0.883	0.844	0.858	0.906	0.885	0.946
0.835	0.885	0.885	0.859	0.874	0.887	0.860	0.907	0.847
0.841	0.183	0.896	0.930	0.184	0.900	0.918	0.913	0.942
0.833	0.803	0.190	0.884	0.933	0.942	0.925	0.199	0.927
0.833	0.913	0.892	0.187	0.907	0.898	0.901	0.890	0.898
0.783	0.846	0.912	0.915	0.915	0.924	0.900	0.882	0.914
0.784	0.857	0.884	0.882	0.900	0.937	0.916	0.930	0.881
0.838	0.859	0.865	0.798	0.817	0.945	0.894	0.911	0.972
0.843	0.901	0.909	0.867	0.915	0.904	0.925	0.899	0.916
0.901	0.877	0.919	0.917	0.863	0.916	0.957	0.959	0.955
0.847	0.817	0.896	0.882	0.838	0.921	0.891	0.890	0.899
0.897	0.899	0.899	0.948	0.858	0.977	0.913	0.944	0.879
0.185	0.886	0.923	0.878	0.912	0.919	0.190	0.913	0.938
0.859	0.833	0.866	0.928	0.891	0.864	0.903	0.930	0.919
0.846	0.188	0.175	0.861	0.807	0.790	0.981	0.964	0.193
0.895	0.919	0.916	0.924	0.874	0.920	0.927	0.921	0.195
0.886	0.821	0.880	0.933	0.917	0.820	0.873	0.808	0.937
0.863	0.836	0.936	0.927	0.893	0.819	0.894	0.903	0.885
0.778	0.747	0.893	0.918	0.897	0.901	0.956	0.893	0.883
0.191	0.885	0.871	0.948	0.893	0.828	0.929	0.878	0.919
0.835	0.870	0.858	0.873	0.873	0.184	0.929	0.915	0.941
0.854	0.884	0.832	0.841	0.838	0.909	0.944	0.913	0.928
0.903	0.837	0.923	0.889	0.915	0.899	0.896	0.931	0.197
0.865	0.892	0.918	0.935	0.727	0.831	0.912	0.895	0.917
0.885	0.844	0.897	0.868	0.910	0.844	0.896	0.891	0.920
0.824	0.895	0.863	0.869	0.905	0.923	0.847	0.950	0.850
0.862	0.796	0.845	0.848	0.879	0.938	0.932	0.904	0.876
0.870	0.905	0.833	0.818	0.878	0.910	0.909	0.825	0.896
0.925	0.833	0.800	0.929	0.879	0.914	0.916	0.950	0.889
0.827	0.846	0.189	0.952	0.977	0.907	0.905	0.934	0.900

Tabla A.4: Resultados de la configuración paramétrica para la instancia I4 con raza objetivo negra.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
$\sum Rango$	72	65	82	118	114	127	170	170	162
P_{Rango}	2.40	2.17	2.73	3.93	3.80	4.23	5.67	5.67	5.40
$Rango$	6.60	6.83	6.27	5.07	5.20	4.766	3.33	3.33	3.60

Tabla A.5: Test de Friedman para I1.

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$
$\sum Rango$	83	73	93	114	103	151	164	145	154
P_{Rango}	2.77	2.43	3.10	3.80	3.43	5.03	5.47	4.83	5.13
$Rango$	6.23	6.57	5.90	5.20	5.57	3.97	3.53	4.17	3.87

Tabla A.6: Test de Friedman para I2.

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$
$\sum Rango$	60	74	60	131	117	157	160	151	170
P_{Rango}	2.00	2.47	2.00	4.37	3.90	5.23	5.33	5.03	5.67
$Rango$	7.00	6.53	7.00	4.63	5.10	3.77	3.67	3.97	3.33

Tabla A.7: Test de Friedman para I3.

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$
$\sum Rango$	60	71	103	132	116	138	163	152	145
P_{Rango}	2.00	2.37	3.43	4.40	3.87	4.60	5.43	5.07	4.83
$Rango$	7.00	6.63	5.57	4.60	5.13	4.40	3.57	3.93	4.17

Tabla A.8: Test de Friedman para I4.

Anexo B

Validación

B.1. Evolución del hipervolumen

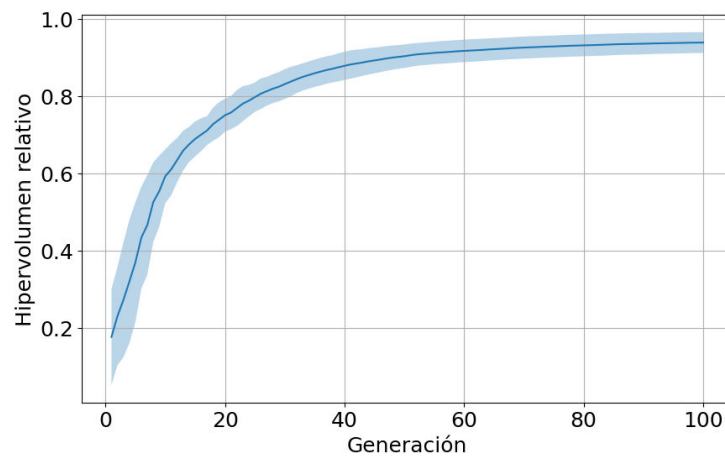


Figura B.1: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I6.

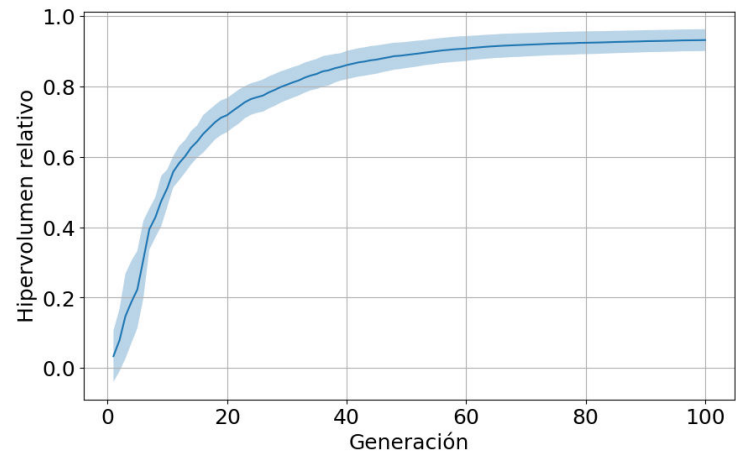


Figura B.2: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I6.

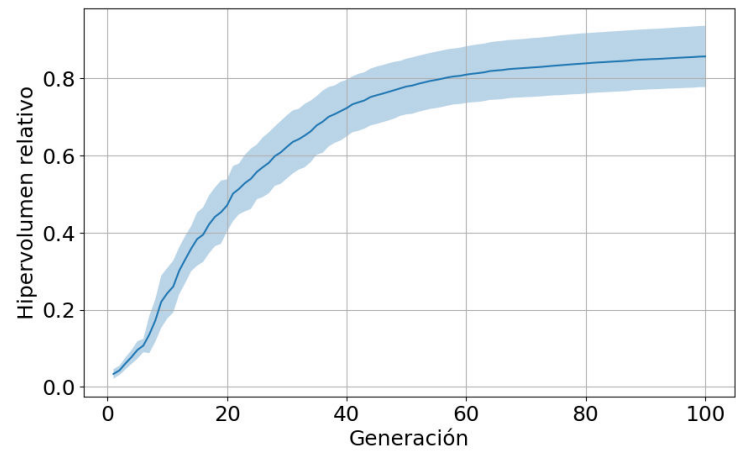


Figura B.3: Evolución de la media y desviación estándar del hipervolumen relativo para la instancia I7.

B.2. Evolución del frente de Pareto

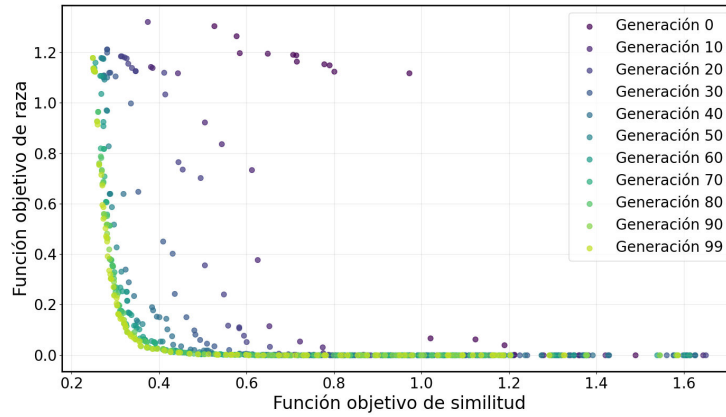


Figura B.4: Evolución del frente de Pareto para la instancia I5.

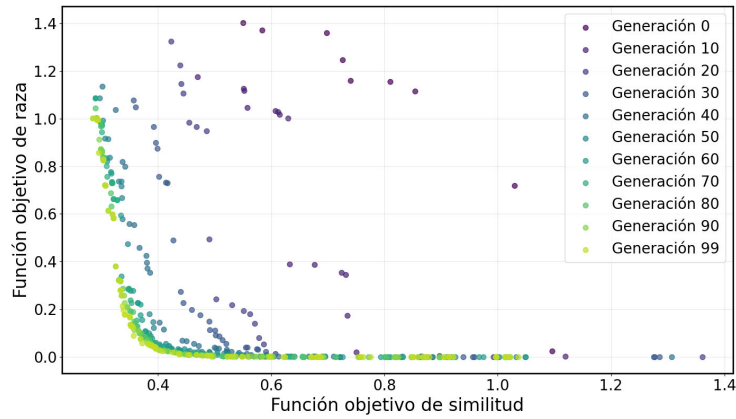


Figura B.5: Evolución del frente de Pareto para la instancia I6.

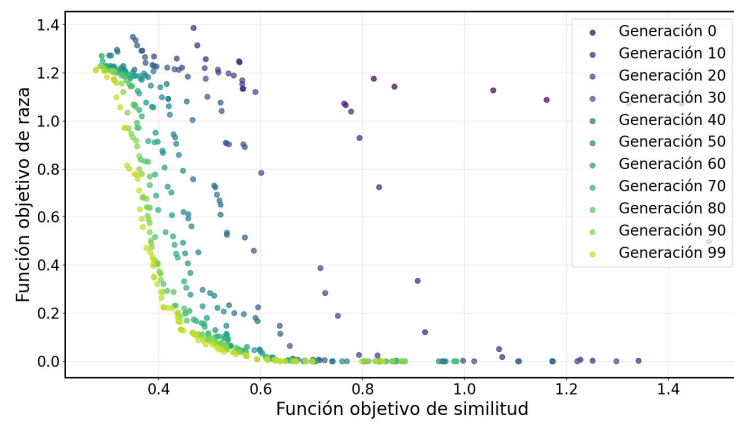


Figura B.6: Evolución del frente de Pareto para la instancia I7.

B.3. Soluciones no dominadas

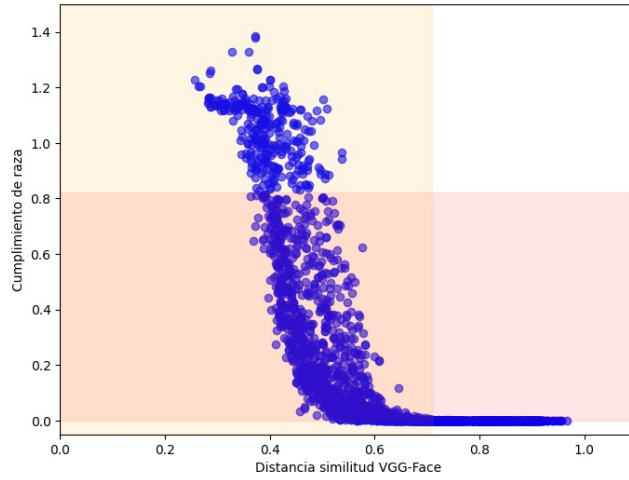


Figura B.7: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I5.

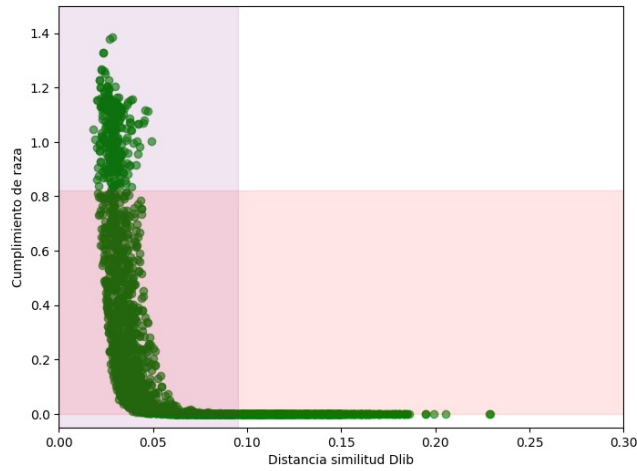


Figura B.8: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I5.

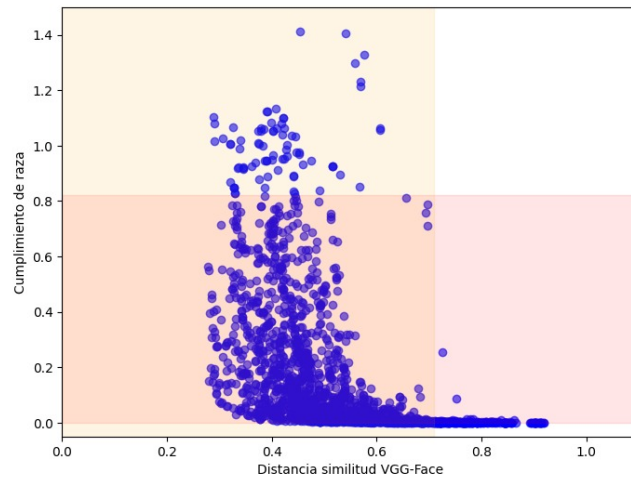


Figura B.9: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I6.

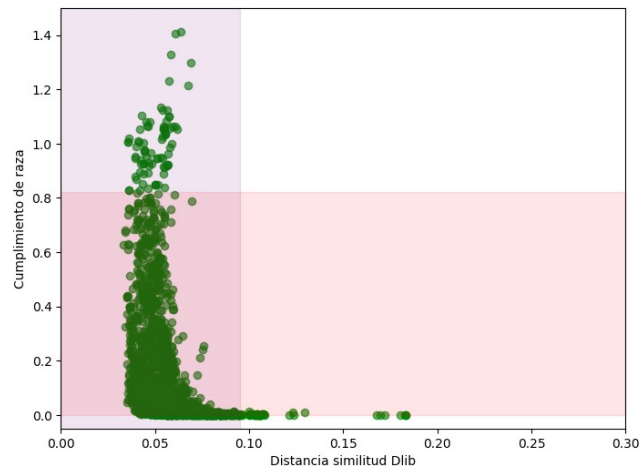


Figura B.10: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I6.

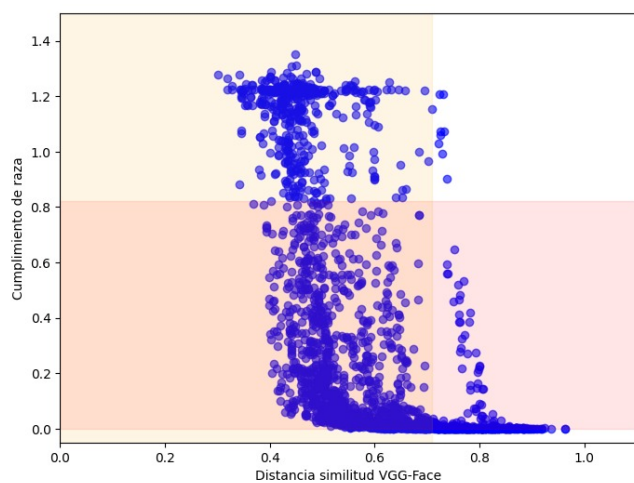


Figura B.11: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I7.

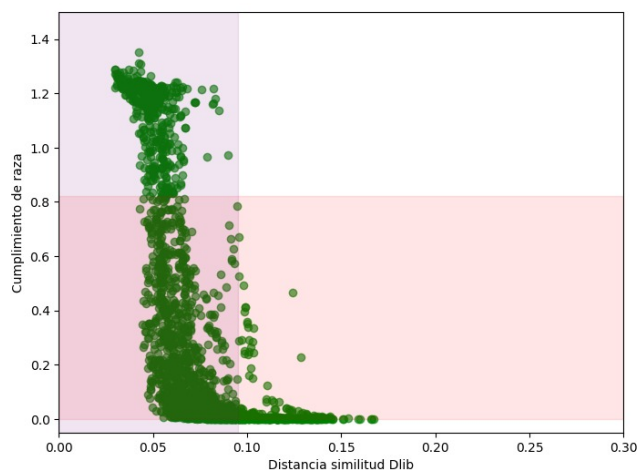


Figura B.12: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I7.

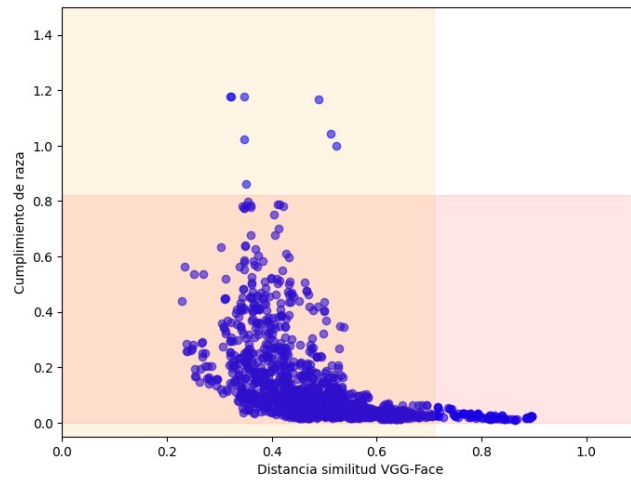


Figura B.13: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I8.

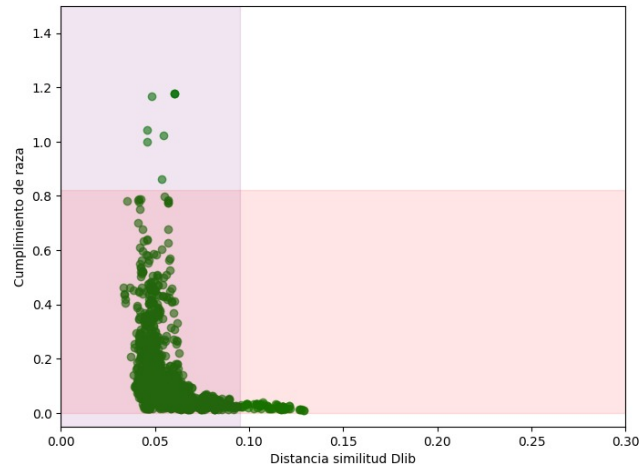


Figura B.14: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I8.

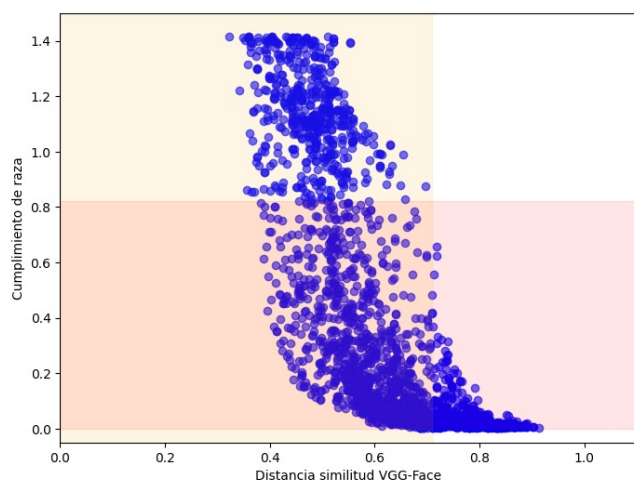


Figura B.15: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I9.

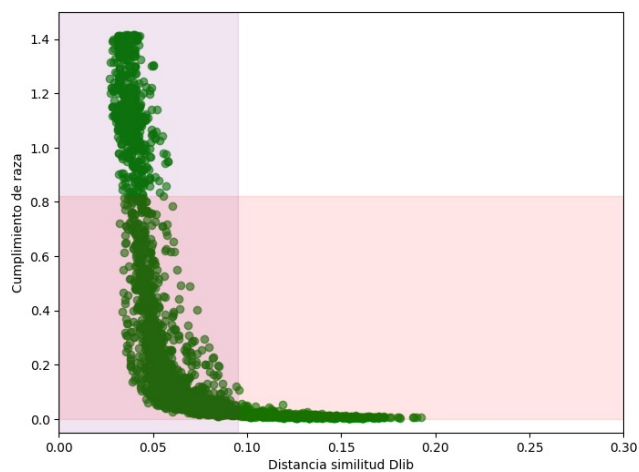


Figura B.16: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I9.

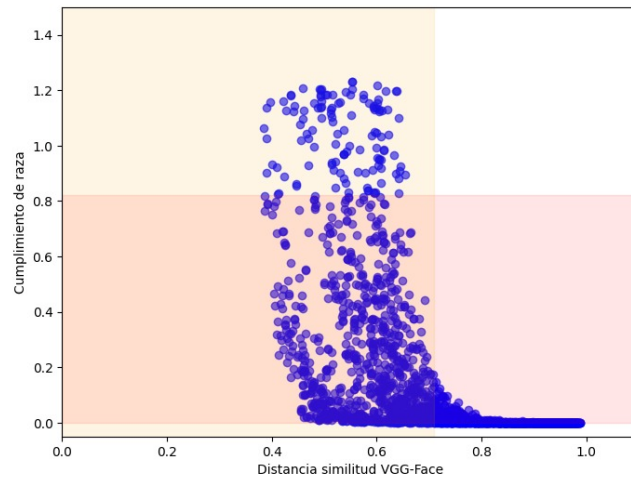


Figura B.17: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I10.

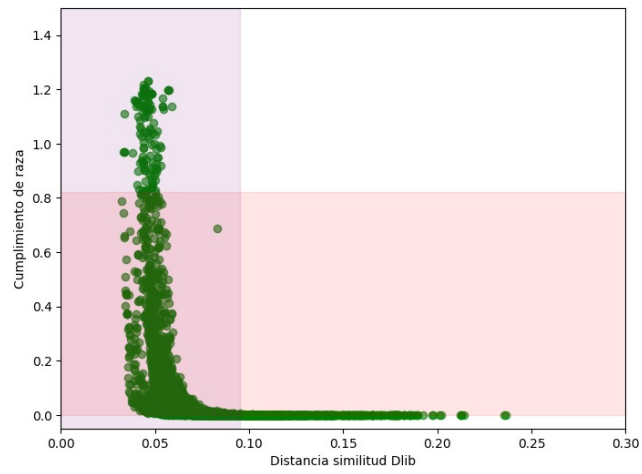


Figura B.18: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I10.

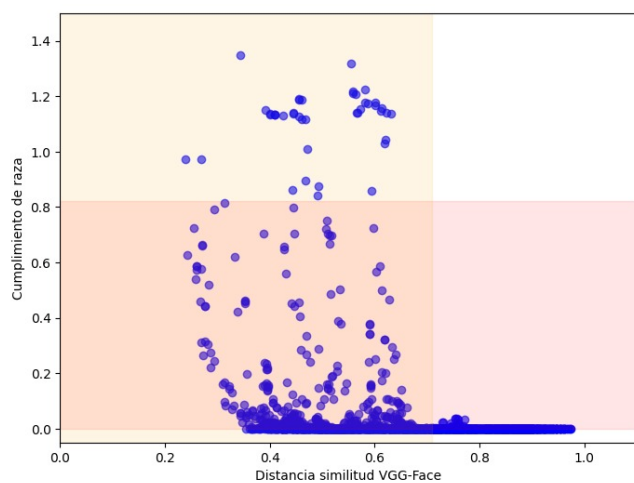


Figura B.19: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I11.

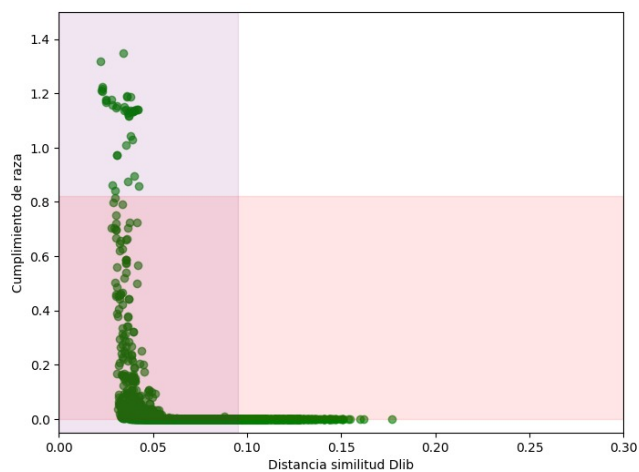


Figura B.20: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I11.

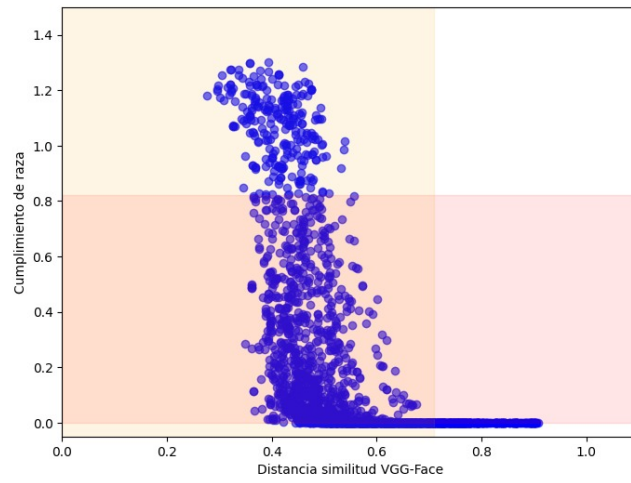


Figura B.21: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I12.

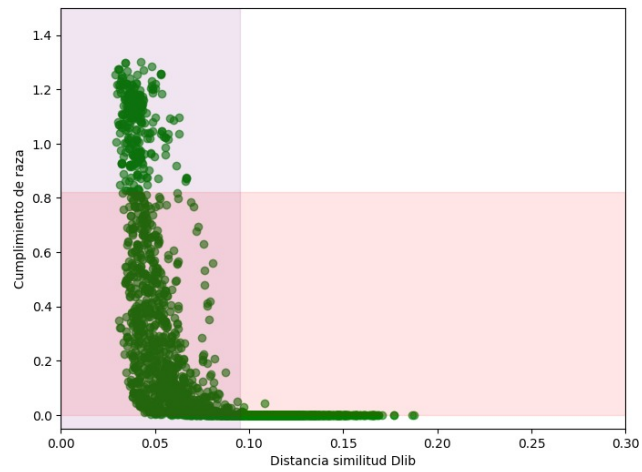


Figura B.22: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I12.

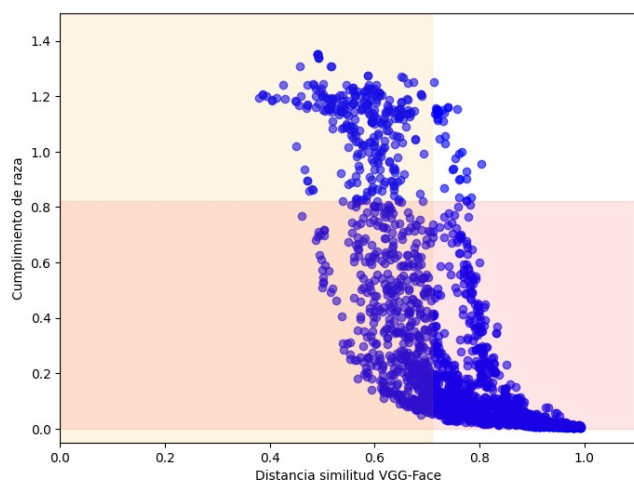


Figura B.23: Soluciones no dominadas ubicadas según el valor de similitud de VGG-Face y de raza para la instancia I13.

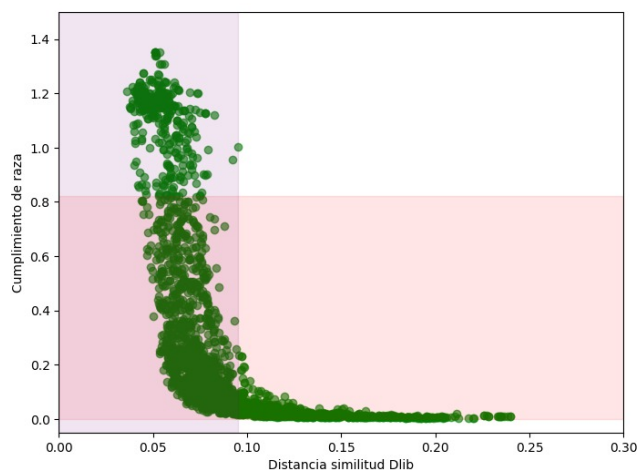


Figura B.24: Soluciones no dominadas ubicadas según el valor de similitud de Dlib y de raza para la instancia I13.

B.4. Calidad de resultados

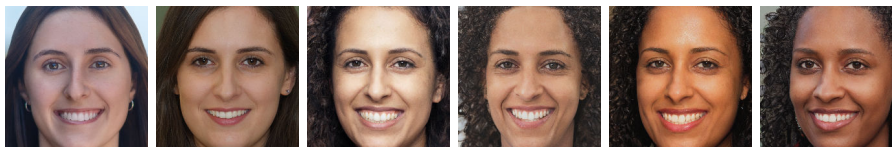


Figura B.25: Instancia I5 e imágenes representativas del frente de Pareto de referencia.

Tabla B.1: Distancias de similitud de las cinco imágenes representativas de la instancia I5.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3284	0.4211	0.4946	0.5188	0.6056
Dlib	0.0264	0.0221	0.0306	0.0361	0.0637

Tabla B.2: Distribución racial de las cinco imágenes representativas de la instancia I5.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.47	0.05	0.01	0	0
Negra	0	0.48	0.87	0.97	1
Latina	0.44	0.30	0.07	0.02	0
Asiático oriental	0	0	0	0	0
Asiático suroriental	0	0	0	0	0
India	0.04	0.15	0.05	0.01	0
Medio oriental	0.05	0.02	0	0	0

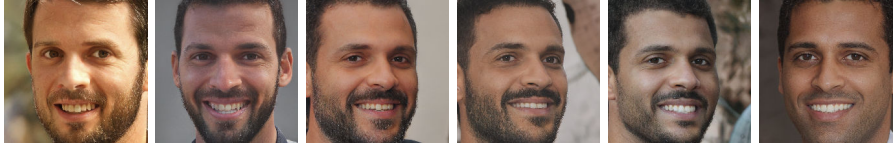


Figura B.26: Instancia I6 e imágenes representativas del frente de Pareto de referencia.

Tabla B.3: Distancias de similitud de las cinco imágenes representativas de la instancia I6.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3250	0.3199	0.4116	0.5055	0.6499
Dlib	0.0358	0.0446	0.0414	0.0433	0.0669

Tabla B.4: Distribución racial de las cinco imágenes representativas de la instancia I6.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.29	0.12	0	0	0
Negra	0	0.03	0.01	0	0
Latina	0.48	0.24	0.02	0.01	0.01
Asiático oriental	0	0	0	0	0
Asiático suroriental	0	0	0	0	0
India	0.16	0.47	0.96	0.99	0.99
Medio oriental	0.07	0.14	0.01	0	0



Figura B.27: Instancia I7 e imágenes representativas del frente de Pareto de referencia.

Tabla B.5: Distancias de similitud de las cinco imágenes representativas de la instancia I7.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3129	0.4178	0.4439	0.6569	0.6276
Dlib	0.0324	0.0471	0.0473	0.0755	0.0920

Tabla B.6: Distribución racial de las cinco imágenes representativas de la instancia I7.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0	0	0	0	0
Negra	0	0.01	0.01	0.01	0
Latina	0	0.18	0.11	0.03	0.01
Asiático oriental	0.76	0.07	0.01	0	0
Asiático suroriental	0.24	0.35	0.28	0.04	0.01
India	0	0.38	0.56	0.92	0.98
Medio oriental	0	0.01	0.01	0	0



Figura B.28: Instancia I8 e imágenes representativas del frente de Pareto de referencia.

Tabla B.7: Distancias de similitud de las cinco imágenes representativas de la instancia I8.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.2443	0.4648	0.6352	0.6175	0.6560
Dlib	0.0431	0.0458	0.0669	0.0745	0.0888

Tabla B.8: Distribución racial de las cinco imágenes representativas de la instancia I8.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.03	0.01	0.02	0.01	0.01
Negra	0	0	0	0	0
Latina	0.53	0.98	0.98	0.97	0.99
Asiático oriental	0.22	0	0	0	0
Asiático suroriental	0.21	0.01	0	0.01	0
India	0	0	0	0.01	0
Medio oriental	0	0	0	0	0



Figura B.29: Instancia I9 e imágenes representativas del frente de Pareto de referencia.

Tabla B.9: Distancias de similitud de las cinco imágenes representativas de la instancia I9.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3890	0.4873	0.6247	0.6230	0.6846
Dlib	0.0327	0.0354	0.0546	0.0768	0.1195

Tabla B.10: Distribución racial de las cinco imágenes representativas de la instancia I9.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0	0	0	0	0
Negra	1.00	0.02	0	0	0
Latina	0	0.16	0	0	0
Asiático oriental	0	0.45	0.78	0.98	1.00
Asiático suroriental	0	0.36	0.22	0.02	0
India	0	0	0	0	0
Medio oriental	0	0	0	0	0



Figura B.30: Instancia I10 e imágenes representativas del frente de Pareto de referencia.

Tabla B.11: Distancias de similitud de las cinco imágenes representativas de la instancia I10.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.5386	0.4373	0.4547	0.8771	0.7841
Dlib	0.0486	0.0441	0.0504	0.0869	0.0702

Tabla B.12: Distribución racial de las cinco imágenes representativas de la instancia I10.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.01	0.01	0.01	0	0
Negra	0.24	0.39	0.66	0.97	1.00
Latina	0.34	0.42	0.22	0.02	0
Asiático oriental	0	0	0	0	0
Asiático suroriental	0	0	0	0	0
India	0.40	0.16	0.10	0.01	0
Medio oriental	0.01	0.02	0	0	0



Figura B.31: Instancia I11 e imágenes representativas del frente de Pareto de referencia.

Tabla B.13: Distancias de similitud de las cinco imágenes representativas de la instancia I11.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.2618	0.4375	0.3595	0.4868	0.4002
Dlib	0.0307	0.0284	0.0328	0.0384	0.0574

Tabla B.14: Distribución racial de las cinco imágenes representativas de la instancia I11.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.27	0.32	0.95	1.00	1.00
Negra	0.01	0	0	0	0
Latina	0.61	0.52	0.03	0.00	0
Asiático oriental	0.01	0.04	0.01	0	0
Asiático suroriental	0.06	0.05	0	0	0
India	0.02	0.04	0	0	0
Medio oriental	0.01	0.03	0.01	0	0



Figura B.32: Instancia I12 e imágenes representativas del frente de Pareto de referencia.

Tabla B.15: Distancias de similitud de las cinco imágenes representativas de la instancia I12.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.3217	0.4800	0.4617	0.4427	0.5476
Dlib	0.0405	0.0372	0.0369	0.0466	0.0778

Tabla B.16: Distribución racial de las cinco imágenes representativas de la instancia I12.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0	0.37	0.82	0.99	1.00
Negra	0.71	0.14	0.01	0	0
Latina	0.02	0.33	0.13	0.01	0
Asiático oriental	0	0	0	0	0
Asiático suroriental	0	0.00	0.00	0	0
India	0.27	0.11	0.01	0	0
Medio oriental	0.00	0.05	0.03	0	0



Figura B.33: Imágenes equidistantes a la instancia I13.

Tabla B.17: Tabla de similitud de las 5 equidistantes I13.

<i>Reconocedor</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
VGG-Face	0.4262	0.5913	0.6155	0.7892	0.9016
Dlib	0.0430	0.0529	0.0537	0.0699	0.1029

Tabla B.18: Distribución racial de las cinco imágenes representativas de la instancia I13.

<i>Categoría racial</i>	<i>Img 1</i>	<i>Img 2</i>	<i>Img 3</i>	<i>Img 4</i>	<i>Img 5</i>
Blanca	0.29	0.21	0.05	0.01	0
Negra	0	0	0	0	0
Latina	0.61	0.36	0.09	0	0
Asiático oriental	0	0.3	0.79	0.96	0.99
Asiático suroriental	0	0.08	0.07	0.03	0.01
India	0.01	0	0	0	0
Medio oriental	0.1	0.05	0.01	0	0

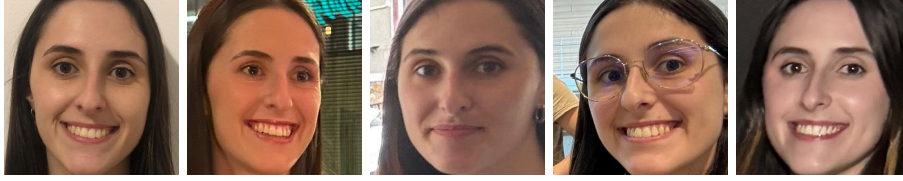


Figura B.34: Imágenes alternativas a la instancia I5.

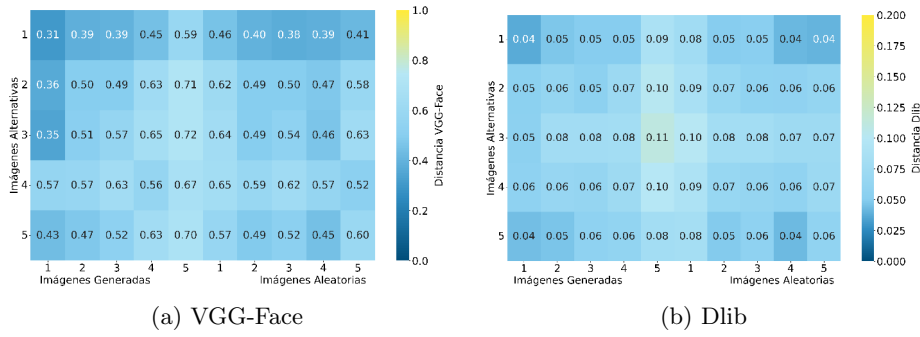


Figura B.35: Heatmaps de distancias de similitud para la instancia I5.



Figura B.36: Imágenes alternativas a la instancia I7.

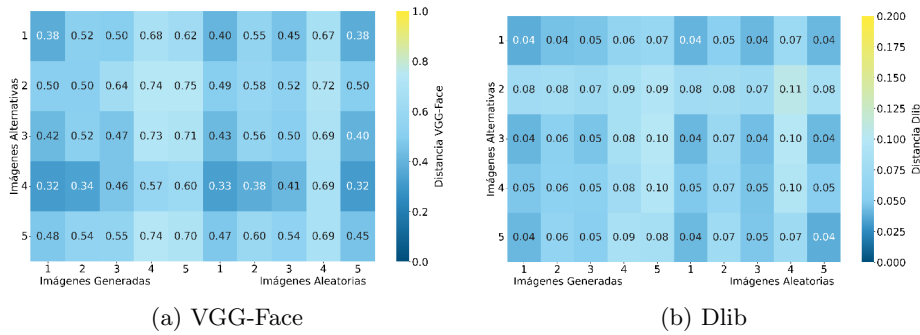


Figura B.37: Heatmaps de distancias de similitud para la instancia I7.

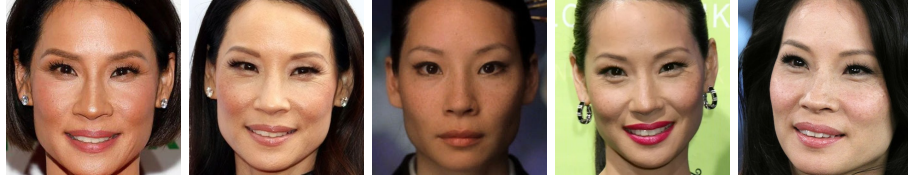


Figura B.38: Imágenes alternativas a la instancia I8.

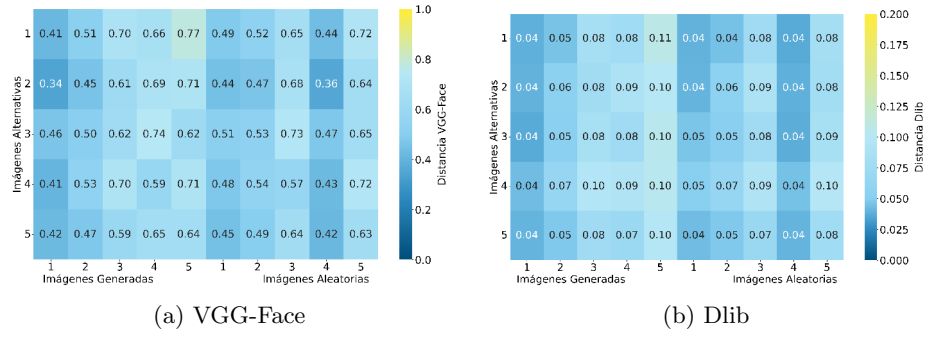


Figura B.39: Heatmaps de distancias de similitud para la instancia I8.



Figura B.40: Imágenes alternativas a la instancia I9.

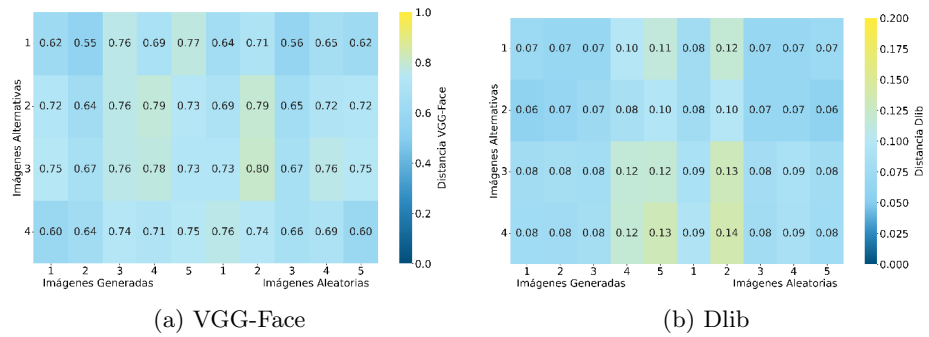


Figura B.41: Heatmaps de distancias de similitud para la instancia I9.



Figura B.42: Imágenes alternativas a la instancia I10.

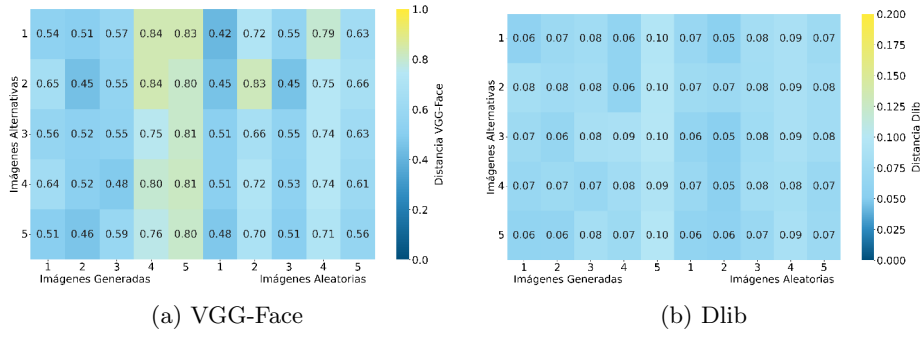


Figura B.43: Heatmaps de distancias de similitud para la instancia I10.

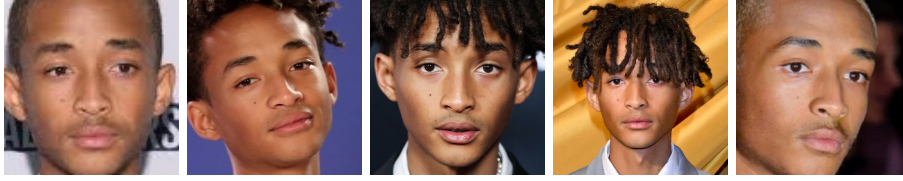


Figura B.44: Imágenes alternativas a la instancia I11.

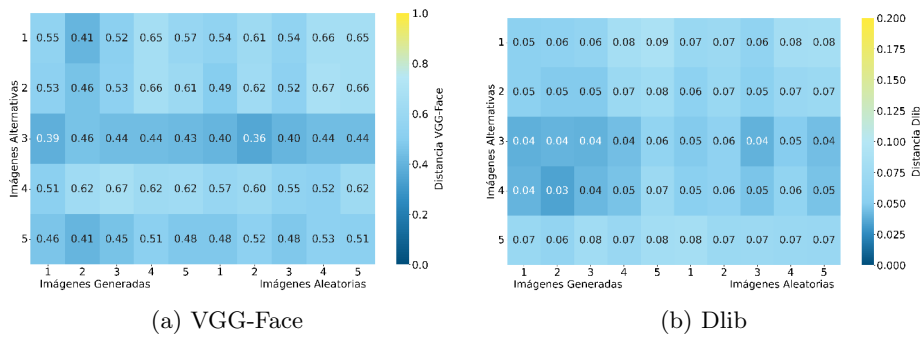


Figura B.45: Heatmaps de distancias de similitud para la instancia I11.

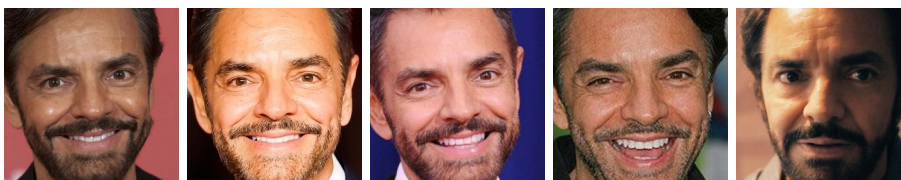


Figura B.46: Imágenes alternativas a la instancia I13.

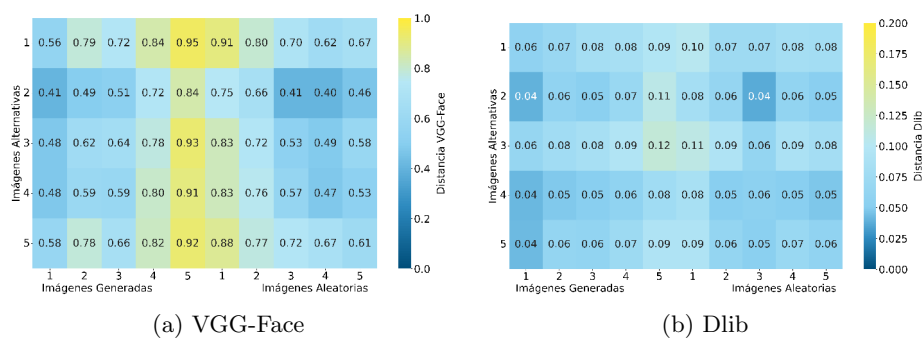


Figura B.47: Heatmaps de distancias de similitud para la instancia I13.