

Generación de imágenes usando redes adversarias generativas para aumentación de datos de entrenamiento para modelos de re-identificación.

Laura Álvarez González

Asesores:

Dr. Víctor Uc Cetina

Dra. Anabel Martín González

Universidad Autónoma de Yucatán

Facultad de Matemáticas

Maestría en Ciencias de la Computación

Resumen En el trabajo se propone el uso de redes adversarias generativas (GAN's) para la aumentación de bases de datos para entrenar y mejorar el desempeño de redes neuronales convolucionales en tareas de re-identificación. La metodología a seguir se basa en el modelo conocido como StyleGAN, la cual es capaz de generar imágenes sintéticas de personas con características específicas deseadas, se denominan estilos. La motivación principal es estudiar la capacidad que tienen las redes GAN's para generar imágenes sintéticas que puedan aumentar el número de imágenes disponibles para entrenar una red convolucional, a partir de un número reducido de ellas. Se utilizará como caso de estudio una red neuronal convolucional para el problema de re-identificación de personas en imágenes obtenidas con múltiples cámaras, ubicadas en lugares diferentes. El problema de re-identificación de personas es actualmente de interés para el desarrollo de sistemas de video vigilancia más precisos.

Palabras clave: Redes Adversarias Generativas · Redes Neuronales Convolucionales · Re-identificación de personas.

Índice general

Tabla de contenidos	I
1. Introducción	1
1.1. Objetivos	3
1.1.1. Objetivos específicos	3
1.2. Organización de la tesis	3
2. Estado del arte	5
2.1. Transferencia de estilo entre distintos dominios	10
2.2. Modificación de la postura	16
2.3. Imágenes artificiales aleatorias	20
2.4. Revisiones del estado del arte	23
3. Marco teórico	25
3.1. Redes neuronales artificiales	26
3.2. Redes generativas adversarias	30
3.2.1. StyleGAN	32
3.3. Modelo de re-identificación	42
4. Metodología	43
4.1. Generación de imágenes artificiales	45
4.2. Modelo de re-identificación	54
5. Resultados experimentales	57
5.1. Generación de imágenes	57
5.2. Re-identificación	72
6. Conclusiones y trabajo a futuro	83

Índice de figuras

1.1. Ejemplo de imágenes obtenidas por las cámaras de seguridad. En verde se marca las imágenes correspondientes a una misma persona.	2
2.1. Transferencia de estilos. A través de una imagen de entrada se generan nuevas imágenes con diferentes estilos.	7
2.2. Modificación de la postura. A través de una imagen de entrada y el mapa de calor de una postura. Se genera una nueva imagen de la persona con la postura dada.	8
2.3. A la izquierda imágenes reales de la base de datos Market- 1501 (1). A la derecha imágenes artificiales generadas con Sty- legan3 (2)	8
2.4. Linea de tiempo del estado del arte. ■ Revisiones del estado del arte. ■■ Transferencia de estilo entre distintos dominios. ■■■ Modificación de la postura. ■■■■ Imágenes artificiales aleatorias.	9
2.5. Transferencia de estilos. Dada una imagen de entrada se mo- difica su estilo, la tonalidad, contraste e iluminación con base a los estilos de las otras cámaras sin que haya un cambio de estructura.	10
2.6. Aplicando CamStyle (3) a 2 dominios. Las imágenes del do- minio A se convierten al dominio B y viceversa.	11
2.7. Ejemplos de imágenes de diferentes bases de datos. Market- 1501 (1), DukeMTMC-ReID (4), CUHK03 (5) y VIPeR (6) . .	12
2.8. Ejemplo DG-GAN (7). Transferencia de estilo, Transfiere la apariencia de la imagen de la izquierda a todas las imágenes de la derecha. Combinando la apariencia y estructura.	13
2.9. Tipos de extracción de la postura. Mediante articulaciones o mapas de calor.	16

2.10. Transformación de la postura mediante una plantilla.	17
2.11. Diferentes tipos de etiquetados. Arriba a la izquierda etiquetado de una imagen real correspondiente a una persona. LSRO etiquetado de manera proporcional, k = número de clases, etiquetado MpRL en función de la similitud y SLSR etiquetado de manera proporcional en función del grupo al que pertenece, pc = distribución en las clases del grupo al que pertenece.	21
3.1. Ejemplo de red neuronal compuesta por una capa de entrada de tres neuronas, una capa oculta y una capa de salida de dos neuronas.	27
3.2. Ejemplo de una neurona compuesta por tres elementos de entrada x y sus respectivos pesos w	27
3.3. Arquitectura GAN genérica.	32
3.4. Representación gráfica de un Generador. Espacio multidimensional donde cada posición representa una imagen.	34
3.5. Ejemplo Generador de 2 dimensiones, a cada coordenada le pertenece una imagen. Al introducir el vector latente $[0,0]$ como salida se obtiene la image superior izquierda	34
3.6. Problema del enredo. De la imagen 1 se hace una interpolación a la imagen 2. a) interpolación sin el problema del enredo. b) interpolación con el problema del enredo. Cómo se puede observar la interpolación a) es mucho más suave y coherente. .	35
3.7. Se modifica el vector latente en las últimas capas de AdaIN  . Se puede observar los cambios en las características suaves, como puede ser el color del pelo, ojos, piel.	37
3.8. Se modifica el vector latente en las capas del medio y últimas de AdaIN  . Se puede observar los cambios en las características medias y suaves. Se modifica la estructura de la cara mientras que la postura es la misma.	38
3.9. Se modifica el vector latente en las primeras capas de AdaIN  . Se puede observar los cambios en las características fuertes, en la estructura de la imagen, mientras que el color de pelo o piel es el mismo.	38
3.10. Izquierda - imágenes reales. Derecha - obtenidas a través del codificador.	40
4.1. Arquitectura - Generación de imágenes artificiales.	44

4.2. Arquitectura - Re-identificación.	44
4.3. StyleGAN. Arquitectura del Generador y del Discriminador (8)	46
4.4. Diagrama - Transferencia de aprendizaje.	46
4.5. Personas generadas de manera artificial con StyleGAN3. Ninguna de estas personas existe.	47
4.6. Imágenes de la base de datos Market-1501.	48
4.7. Partiendo del vector latente inicial se generan nuevas imágenes. Para mezclar los estilos en las capas se le introduce el vector latente de las otras imágenes mientras que las capas blancas se le introduce el vector latente de la imagen original.	50
4.8. Ejemplo de interpolación desde una imagen de entrada hasta la misma image envejecida.	50
4.9. Entrenamiento del codificador de StyleGAN3.	51
5.1. Evolución del rendimiento del modelo mediante la métrica FID en diferentes épocas del entrenamiento de StyleGAN3. . .	59
5.2. Imágenes artificiales generadas después del entrenamiento de manera aleatoria.	60
5.3. Aplicación de YoloV4 tiny sobre las imágenes de la base de datos Market-1501 utilizando diferentes número de umbrales. El porcentaje de error son las imágenes que no ha clasificado como peatón.	62
5.4. Histograma de la métrica SSIM (9) sobre la base de datos Market-1501. Número de imágenes que han obtenido el mismo valor SSIM. Se seleccionó una imagen por persona y se comparó con el resto de imágenes de esa misma persona. Cómo se puede observar la mayoría de las imágenes rondan el umbral 0.75 en adelante.	63
5.5. A) Imágenes reales de la base de datos Market-1501. B) Imágenes artificiales.	64
5.6. Semilla - es la imagen generada de manera aleatoria a la cual se le van a modificar sus vectores latentes para modificar sus características medias. Generadas - son las imágenes que se han generado al modificar los vectores latentes de la imagen semilla.	65
5.7. Ejemplo de algunas imágenes descartadas utilizando el modelo Yolo V4 tiny para detectar peatones.	66

5.8. Ejemplo de algunas imágenes descartadas utilizando la métrica SSIM (9). Partiendo como base de una de las imágenes de una persona (imagen original), se compara con el resto de imágenes generadas de esa misma persona.	66
5.9. LPIPS	68
5.10. L2	68
5.11. MOCO	68
5.12. LPIPS	69
5.13. L2	69
5.14. MOCO	69
5.15. Pares de imágenes. Derecha se encuentra la imagen real, y a la izquierda su homólogo obtenido a través del codificador dentro del espacio latente de StyleGAN3. Se puede observar que el modelo funciona mejor cuando se muestra el cuerpo completo, aunque no alcanza la calidad de las imágenes generadas de manera aleatoria.	70
5.16. Semilla - imagen real. Generadas - son las imágenes que se han generado al modificar los vectores latentes de la imagen semilla.	71
5.17. Base, sin personas añadidas. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	73
5.18. Añadiendo 100 personas. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	73
5.19. Añadiendo 200 personas. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	74
5.20. Añadiendo 320 personas. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	74
5.21. Rendimiento de algunos modelos entrenados con diferente número de personas generadas de manera artificial (ver Tabla 5.10). Añadiendo 0, 40, 80, 120, 160, 200, 240, 280 y 320 personas . .	76
5.22. Resultados modelo de re-identificación. Comparación de resultados con el modelo base () y modelo despues de añadir 280 personas (). Query es la imagen de la persona a buscar, y las siguiente imágenes representan la salida del modelo, el color verde es un acierto y rojo un error.	77

5.23. Base, sin personas añadidas. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	79
5.24. Añadiendo 35 imágenes a cada persona. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	79
5.25. Añadiendo 100 imágenes a cada persona. Izquierda , función de pérdida durante el entrenamiento y validación. Derecha , porcentaje de error en Rank1 durante el entrenamiento y validación.	80
5.26. Rendimiento de algunos modelos entrenados con diferente número de imágenes artificiales por persona (ver Tabla 5.11). Añadiendo 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 y 120 imágenes por persona.	81

Capítulo 1

Introducción

La re-identificación de personas es una técnica utilizada en el campo de la inteligencia artificial y el aprendizaje automático para reconocer a una persona en diferentes imágenes o videos, incluso si estos presentan diferentes ángulos, iluminación o vestimenta. Esta técnica se emplea en diversas aplicaciones, como la vigilancia de seguridad, la identificación de personas en imágenes digitales y el análisis de comportamientos en videos. Se basa en el uso de modelos de aprendizaje automático que aprenden a reconocer las características que identifican a una persona en diferentes imágenes o videos. Estos modelos se entrena con un conjunto de datos que contiene imágenes o videos de personas, junto con información sobre las características que identifican a cada persona. La re-identificación de personas puede ser un desafío debido a la variabilidad de las características que identifican a una persona, como la ropa, peinado y otros aspectos que pueden cambiar su apariencia. Asimismo, puede ser un desafío si se cuenta con un número limitado de datos de entrenamiento debido a la privacidad de las personas que aparecen en las imágenes o videos. Para superar estos desafíos, se pueden utilizar técnicas de pre-procesamiento de imágenes y videos, así como técnicas de aprendizaje profundo que permiten al modelo adaptarse a las variaciones en la apariencia de una persona y reconocer características relevantes en imágenes o videos de baja calidad.

En la actualidad, los conjuntos de datos de entrenamiento más destacados para la re-identificación de personas están muy limitados debido a que no contienen un gran número de imágenes. Por ejemplo, Market1501 incluye solamente 1501 personas grabadas con 6 cámaras diferentes, mientras que DukeMTMC-reID cuenta con 702 personas en 8 cámaras diferentes. Tal co-

mo se puede apreciar en la Fig. 1.1, se presentan diversos desafíos para la re-identificación de personas en imágenes, como baja resolución de imágenes, variaciones en la iluminación y el contraste, así como otros factores que complican la tarea como cambios en la ropa, presencia de objetos como mochilas o suéteres, y la presencia de obstáculos o personas en el fondo que limitan la visibilidad de la persona de interés en un espacio abierto.



Figura 1.1: Ejemplo de imágenes obtenidas por las cámaras de seguridad. En verde se marca las imágenes correspondientes a una misma persona.

En este estudio se investiga la utilización de una red generativa adversaria, junto con técnicas de aumento de datos, para entrenar un modelo de re-identificación de personas. La red generativa adversaria es un tipo de modelo de aprendizaje automático que se emplea para generar imágenes o videos sintéticos que se pueden utilizar como datos de entrenamiento adicionales. Se analizan diversas técnicas para ampliar los datos de entrenamiento, tales como la generación de imágenes y la expansión de características, y se evalúa su efectividad en la formación de un modelo de re-identificación de personas mediante el uso de una red generativa adversaria. Además, se comparan los resultados obtenidos con un modelo formado con un conjunto de datos no ampliado.

1.1. Objetivos

El objetivo general de esta investigación es la generación de imágenes artificiales de personas a partir de un conjunto reducido de imágenes de entrenamiento para mejorar el rendimiento en modelos de re-identificación.

1.1.1. Objetivos específicos

Los objetivos específicos son los siguientes:

1. Investigar las técnicas de aumentación de datos para entrenar un modelo de re-identificación de personas.
2. Analizar el uso de una red generativa adversaria para generar imágenes sintéticas que se puedan utilizar como datos de entrenamiento adicionales para un modelo de re-identificación de personas.
3. Evaluar la efectividad de las técnicas de aumentación de datos y de la red generativa adversaria en el entrenamiento de un modelo de re-identificación de personas.
4. Contribuir al desarrollo de técnicas de re-identificación de personas y proporcionar una base para futuras investigaciones en esta área.

1.2. Organización de la tesis

- Capítulo 1 - Introducción

El capítulo introducción de esta tesis tiene como objetivo presentar el contexto y el objetivo general de la investigación. En primer lugar, se presenta el contexto en el que se desarrolla la tesis, incluyendo una breve descripción del campo de la re-identificación de personas y su importancia en aplicaciones como la vigilancia de seguridad y el análisis de comportamientos en videos.

- Capítulo 2 - Estado del arte

El objetivo de este capítulo es presentar una revisión de la literatura relevante en el campo de la re-identificación de personas y la utilización de redes generativas adversarias para aumentar datos de entrenamiento.

- Capítulo 3 - Marco teórico

En este capítulo se presenta el marco teórico en el que se desarrolla la investigación. En primer lugar, se presenta una revisión de los conceptos básicos sobre redes generativas adversarias, incluyendo una descripción de cómo estas redes se utilizan para generar imágenes sintéticas que se pueden utilizar como datos de entrenamiento adicionales. En segundo lugar, se presenta una revisión de los conceptos básicos en el campo de la re-identificación de personas, incluyendo una descripción de las técnicas utilizadas para reconocer a una persona en diferentes imágenes o videos, así como los desafíos que se enfrentan en esta tarea.

- Capítulo 4 - Metodología

Se precisa la metodología de análisis de datos utilizada, incluyendo las técnicas de aumentación de datos utilizadas, así como la metodología de evaluación del desempeño de un modelo de re-identificación de personas.

- Capítulo 5 - Resultados experimentales

En primer lugar, se presentan los resultados de la aplicación de técnicas de aumento de datos en el conjunto de datos utilizado en la investigación, incluyendo la cantidad de datos aumentados generados. En segundo lugar, se presentan los resultados de la evaluación del desempeño de un modelo de re-identificación de personas entrenado con datos aumentados mediante una red generativa adversaria. En tercer lugar, se comparan los resultados obtenidos con un modelo entrenado con un conjunto de datos sin aumento, y se analizan las diferencias en el desempeño entre ambos modelos..

- Capítulo 6 - Conclusiones y trabajo a futuro

Tiene como objetivo presentar las conclusiones obtenidas en la investigación y proponer líneas de trabajo futuro. Se presentan las conclusiones generales de la investigación, incluyendo un resumen de los resultados obtenidos y una discusión de su significado y relevancia en el campo de la re-identificación de personas y la utilización de redes generativas adversarias para aumentar datos de entrenamiento y las limitaciones de la investigación y se proponen líneas de trabajo futuro para superar estas limitaciones y continuar el desarrollo de la investigación en esta área.

Capítulo 2

Estado del arte

El artículo “Generative Adversarial Networks” del año 2014 de Ian Goodfellow *et al.* (10) presenta una nueva clase de redes neuronales denominadas redes generativas adversarias (GANs). Estas redes son un tipo de modelo de aprendizaje automático que se utilizan para generar imágenes o videos sintéticos a partir de un conjunto de datos. Las GANs están formadas por dos redes neuronales entrenadas de manera simultánea, una generadora y una discriminadora. La red generadora se encarga de generar imágenes o videos sintéticos, mientras que la red discriminadora se encarga de evaluar la calidad de las imágenes o videos generados por la red generadora. El propósito de las redes generativas adversarias consiste en que la red generadora pueda producir imágenes o videos con un alto grado de similitud a los objetos y escenarios reales, de tal forma que la red discriminadora no logre diferenciar entre las imágenes o videos auténticos y aquellos generados de manera sintética por la red generadora. El artículo presenta experimentos que demuestran la capacidad de las GANs para generar imágenes o videos sintéticos de alta calidad, y se discute el potencial de estas redes en aplicaciones como la generación de imágenes en 3D, la mejora de la calidad de las imágenes o videos y el análisis de comportamientos en videos.

Desde ese primer artículo se han desarrollado nuevas arquitecturas que mejoran la calidad de las imágenes generadas, como es el caso de la arquitectura propuesta en el año 2017, CycleGan (11) es capaz de mejorar la calidad de las imágenes generadas, logrando transferir el estilo o dominio de un grupo de imágenes a otro grupo uniendo dos redes generativas adversarias. Debido a la mejora en el rendimiento de las redes generativas adversarias se comienza a utilizar para aumentar los datos de entrenamiento que posibilitan la

mejora del rendimiento en modelos de aprendizaje máquina donde las bases de datos están limitadas. En este caso se puede observar en la Fig. 2.4 que a partir del año 2018 hay un incremento notable en el estudio del uso de redes generativas adversas para aumentar los datos de entrenamiento en modelos de re-identificación.

Los artículos han sido agrupados en cuatro categorías, correspondientes a diferentes métodos utilizados para la generación de nuevas imágenes artificiales.

- Transferencia de estilos entre distintos dominios

Mediante una imagen real de entrada, se pueden generar imágenes artificiales utilizando los diferentes estilos, o dominios, con los que el modelo ha sido entrenado. Esto posibilita la transferencia del estilo de un conjunto de datos a otro, como, por ejemplo, la transferencia del estilo de una pintura a una fotografía real. En las imágenes generadas, se pueden observar cambios con respecto a la imagen original, tales como colores, tonalidades, iluminación, entre otros, pero no hay un cambio en la estructura de la imagen (ver Fig. 2.1).

- Modificación de la postura

El objetivo de estas redes es lograr que la red generadora produzca imágenes de personas con posturas diversas, las cuales sean indistinguibles de las imágenes reales del conjunto de datos utilizado para su entrenamiento. Se utiliza como entrada una imagen de una persona real y un mapa de calor o de articulaciones que corresponde al esqueleto de una postura diferente. De esta forma, se consigue ampliar el conjunto de datos de entrenamiento de un modelo de re-identificación de personas, añadiendo imágenes con posturas variadas de las mismas (ver Fig. 2.2).

- Imágenes artificiales aleatorias

Se generan imágenes artificiales de manera aleatoria y les aplican técnicas de etiquetado (ver Fig. 2.3).

- Revisión del estado del arte

Los escritos referidos al estado del arte son empleados por investigadores y profesionales con el propósito de obtener una panorámica general

de un campo de estudio y para detectar las principales tendencias y desafíos en dicho ámbito. También son utilizados como base para el diseño de nuevas investigaciones y proyectos. En el presente caso, nos centramos en los artículos sobre el estado del arte que se enfocan en el tema específico del uso de redes generativas adversarias con el fin de aumentar los datos de entrenamiento en modelos de re-identificación.

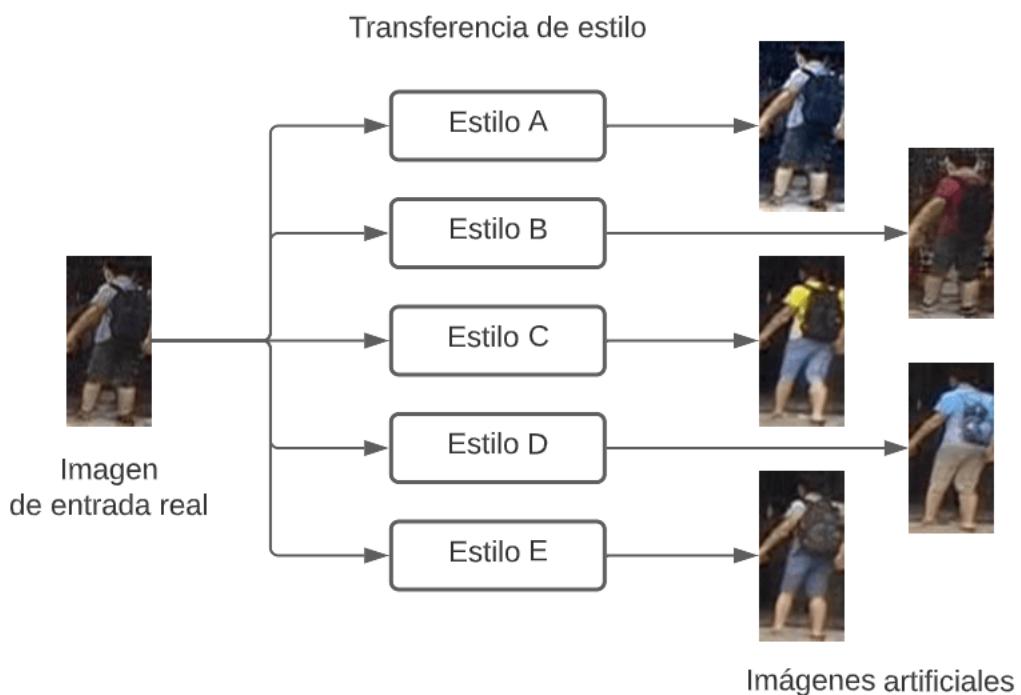


Figura 2.1: Transferencia de estilos. A través de una imagen de entrada se generan nuevas imágenes con diferentes estilos.

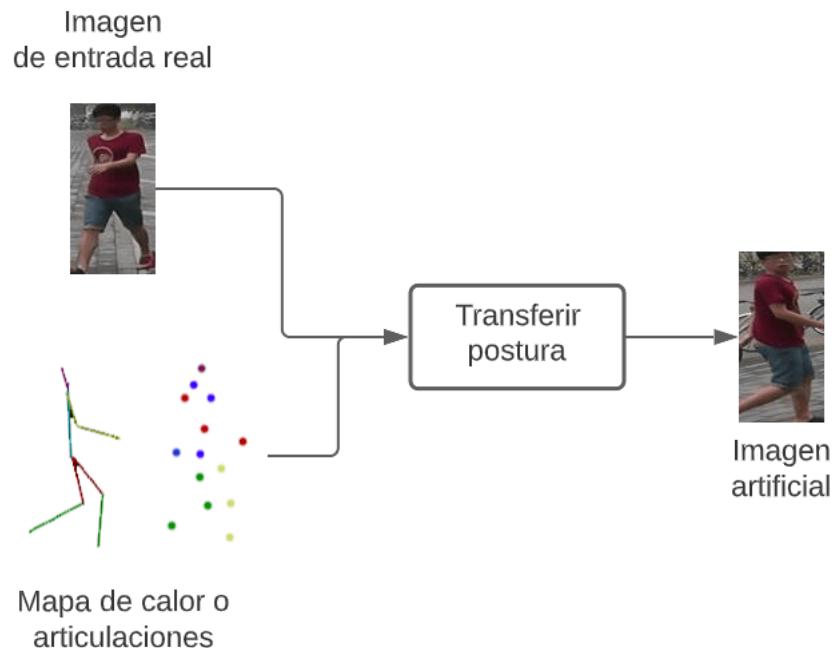


Figura 2.2: Modificación de la postura. A través de una imagen de entrada y el mapa de calor de una postura. Se genera una nueva imagen de la persona con la postura dada.



Figura 2.3: A la izquierda imágenes reales de la base de datos Market-1501 (1). A la derecha imágenes artificiales generadas con Stylegan3 (2)

Review of GAN-Based Person Re-Id		Exploring the Quality of GAN Generated Images for Person Re-Id	Cross-Domain Person Re-Identification Based on Feature Fusion
2021	2020	An Introduction to Person Re-Identification with Generative Adversarial Networks	CGAN-TM: A Novel Domain-to-Domain Transferring Method for Person Re-Id
		Semi-supervised person re-identification by similarity-embedded cycle GANs	A Novel Method for Person Re-Identification: Conditional Network Based on GANs
		Multi-camera GAN for person re-id	Cross-domain person re-id by hybrid supervised and unsupervised learning
2021	2020	Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Id	Unity Style Transfer for Person Re-Identification
2019	2018	CamStyle: A Novel Data Augmentation Method for Person Re-Id	Modality-aware Collaborative Learning for Visible Thermal Person Re-Id
2018	2017	CamStyle: A Novel Data Augmentation Method for Person Re-Id	CamStyle: A Novel Data Augmentation Method for Person Re-Id
2017	2017	Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Id	Joint Discriminative and Generative Learning for Person Re-Id
2018	2017	CamStyle: A Novel Data Augmentation Method for Person Re-Id	Color-Sensitive Person Re-Id
2019	2017	CamStyle: A Novel Data Augmentation Method for Person Re-Id	Cross-Modality Person Re-Id with Generative Adversarial Training
2020	2019	PAC-GAN: An effective pose augmentation scheme for unsupervised cross-view person re-id	CamStyle: A Novel Data Augmentation Method for Person Re-Id
2021	2020	FD-GAN: Pose-guided Feature Distilling GAN for Robust Person Re-Id	Unlabeled Samples Generated by GAN Improve the Person Re-Identification Baseline In-vitro
2021	2019	Instance-guided context rendering for cross-domain person re-identification	Sparse Label Smoothing Regularization for Person Re-Id
2020	2018	A Unified Generative Adversarial Framework for Image Generation and Person Re-Id	Learning Disentangled Representation for Robust Person Re-Identification
2019	2018	GAN-based Pose-aware Regulation for Video-based Person Re-Id	Multi-pseudo Regularized Label for Generated Data In Person Re-Id
2021	2019	PGAN: Part-based nondirect coupling embedded gan for person reid	Real-World Person Re-Identification via SuperResolution and Semi-Supervised Methods
2020	2018	Joint Generative and Contrastive Learning for Unsupervised Person Re-Identification	StyleGAN+LSRG Method for Person Re-Identification
2021	2018	Affine Transform for Skew Correction Based on Generative Adversarial Network Method for Multi-Camera Person Re-Identification	

Figura 2.4: Línea de tiempo del estado del arte.

■ Revisões del estado del arte. ■ Transferencia de estilo entre distintos dominios. ■ Modificación de la postura. ■ Imágenes artificiales aleatorias.

2.1. Transferencia de estilo entre distintos dominios

Las imágenes obtenidas por las cámaras suelen tener diferente resolución o estar en diferentes posiciones. Esto puede provocar que varie la iluminación y la tonalidad entre otros aspectos. Una forma de generar nuevos datos es mediante la transferencia de un dominio a otro o adaptación del dominio, la cual se basa en la idea de transferir el estilo de unas imágenes o imagen a otras (Fig. 2.5) sin que haya una modificación en la estructura o fondo de la imagen. Lo que significa que no hay un cambio en las posiciones de los píxeles quedando las imágenes exactamente iguales.

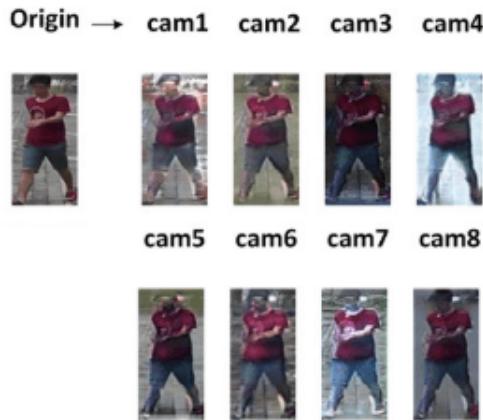


Figura 2.5: Transferencia de estilos. Dada una imagen de entrada se modifica su estilo, la tonalidad, contraste e iluminación con base a los estilos de las otras cámaras sin que haya un cambio de estructura.

Dentro de los trabajos referentes a la transferencia de estilo nos encontramos dentro de la literatura varias maneras de afrontar el problema. En el año 2017 aparece la arquitectura de una red generativa adversaria CycleGAN (11) la cual es capaz de aprender el estilo de unas imágenes y transferirlo a otras imágenes distintas, es decir, transferir el estilo de un dominio a otro. Esto fue un hito dentro de la redes generativas adversarias y a partir del año 2018 empezaron a aparecer gran cantidad de trabajos donde se utilizaba esta arquitectura, en este caso transferir el estilo de una cámara de seguridad a la de otra cámara, como es el trabajo de Zhun Zhong *et al.*, 2018 (3) donde

2.1. TRANSFERENCIA DE ESTILO ENTRE DISTINTOS DOMINIOS11

proponen CamStyle, un método que utiliza la arquitectura de la red generativa adversaria CycleGAN (11). Esta sólo puede transferir el estilo entre dos dominios limitando la arquitectura de tal forma que es necesario generar un modelo por cada par de cámaras de seguridad.



Figura 2.6: Aplicando CamStyle (3) a 2 dominios. Las imágenes del dominio A se convierten al dominio B y viceversa.

Siguiendo la misma filosofía de transferir un estilo de otro en el año 2018 Pingyang Dai *et al.* (12) proponen el modelo cmGAN enfocado en la transferencia de estilos para convertir imágenes de cámaras RGB e infrarrojas. Es el primer artículo donde utilizan la base de datos RGB-Infrared Cross-Modality Re-ID Dataset (13) la cual incluye imágenes de cuatro cámaras infrarrojas y dos RGB. En este caso el discriminador de la red generativa adversa forma parte del extractor de características del modelo de re-identificación. La entrada en el modelo de re-identificación será una imagen infrarroja y deberá buscar a esa persona dentro de las imágenes RGB. En los anteriores artículos nos encontramos con la limitación de transferir un estilo de un dominio A a un dominio B, teniendo la necesidad de duplicar el proyecto por cada diferente estilo, esto añadido a uno de los mayores retos a los que se enfrentan los modelos de re-identificación el cual es el bajo rendimiento que se obtiene cuando en las pruebas del modelo de re-identificación se utilizan imágenes de otra base de datos. Con base a estas problemáticas algunos trabajos proponen la transferencia de dominios entre distintas bases de datos y/o multiples dominios (Fig. 2.7).

En el año 2018 cómo mejora de la arquitectura Camstyle y buscando un



Figura 2.7: Ejemplos de imágenes de diferentes bases de datos. Market-1501 (1), DukeMTMC-ReID (4), CUHK03 (5) y VIPeR (6)

mejor rendimiento utilizando el modelo en distintas bases de datos, se propone la arquitectura M2M-GAN (14) la cual clasifica las imágenes de cada base de datos en subdominios, es decir, por cada una de las cámaras. Pudiendo transferir el sub-dominio del dominio A hacia un sub-dominio del dominio B, siendo el entrenamiento realizado de manera supervisada y requiriendo que todos los datos de las diferentes bases de datos hayan sido etiquetados a mano. Por otro lado en el trabajo de Weijian Deng *et al.* (15), se intenta globalizar el modelo desarrollando la arquitectura SPGAN, transfiriendo las imágenes del estilo de una base de datos a otra, la cual se entrena de manera no supervisada y está compuesta por una red siamesa (16) (SiaNet) y una CycleGan (11). De manera similar Shuren Zhou *et al.* (17) proponen la arquitectura CTGAN. En este trabajo se disminuye la complejidad haciendo que la transferencia de estilos sea de un dominio a varios dominios de otra base de datos utilizando sólo un modelo generador y discriminador. En este caso, usando la arquitectura de una red generativa advesaria StarGAN (18).

Siguiendo esta linea En el año 2020 presentan la arquitectura CGAN-TM (19) propuesta por Yingzhi Tang *et al.* donde convierten las imágenes de una base de datos a otra utilizando como red generativa adversaria una Cyclegan (11). La innovación en este trabajo es la utilización del Self-Labeled

2.1. TRANSFERENCIA DE ESTILO ENTRE DISTINTOS DOMINIOS13

Triplet Net la cual etiqueta las imágenes artificiales generadas para entrenar de manera no supervisada el modelo de re-identificación.

Durante ese mismo año, Yacine Khraimeche *et al.* (20) presentaron la técnica UD-GAN, la cual busca mejorar el rendimiento del modelo de re-identificación mediante el entrenamiento con una base de datos que contiene datos etiquetados, y su posterior evaluación en una base de datos diferente que carece de datos etiquetados.

A partir del año 2019 empiezan a aparecer diferentes arquitecturas más sofisticadas como la propuesta por Zhedong Zheng *et al.* (7) DG-NET utiliza dos codificadores que son capaces de extraer los colores de la imagen, apariencia, y transfieren esos colores a otra imagen en donde se ha extraido la estructura de la persona (Fig. 2.8). Este modelo también utiliza la propia arquitectura de la red generativa adversaria como modelo de re-identificación.



Figura 2.8: Ejemplo DG-GAN (7). Transferencia de estilo, Transfiere la apariencia de la imagen de la izquierda a todas las imágenes de la derecha. Combinando la apariencia y estructura.

A principios del año 2020, Rui Sun *et al.* (21) propuso la arquitectura cTransNet, utilizando como base la red generativa adversaria StarGAN (18). El objetivo de esta arquitectura es desarrollar un solo generador capaz de generar múltiples imágenes de una imagen de entrada, con los diferentes estilos de cada una de las cámaras. Por otro lado, el equipo de Yang Yang *et al.* (22) propuso en su trabajo “*Color-Sensitive Person Re-Identification*” la arquitectura Color Translation GAN (CTGAN), la cual se enfoca en distinguir entre los distintos colores de la ropa y mantener la coherencia de identidad de la persona en relación con el color de su ropa. CTGAN es capaz de identi-

ficar y modificar los colores de la ropa superior (como sudaderas o camisetas) y de la ropa inferior (como pantalones).

Chong Liu *et al.* (23) propuso UnityGAN, la diferencia con respecto a los anteriores trabajos es que genera imágenes artificiales en un estilo que es la combinación de todos los demás estilos, no necesita aprender a transferir por pares. Elimina las diferencias entre estilos dejando un estilo único. La arquitectura utiliza como base DiscoGAN (24) y CycleGAN (11).

En los últimos años, han aparecido modelos semi supervisados y no supervisados debido a la problemática de los modelos de re-identificación, en los cuales se requiere un gran número de imágenes etiquetadas. En el año 2020, Xinyu Zhang *et al.* (25) propuso SECGAN, similarity-embedded CycleGANs (11). Debido a la limitación de datos etiquetados, este método semi supervisado entrena con datos etiquetados y no etiquetados de manera alterna y utiliza ambos codificadores de cada una de las cámaras, A y B, como extractor de características discriminatorias.

En los siguientes trabajos se hace uso de imágenes etiquetadas para transferir su estilo a un dominio sin etiquetar. En el año 2021, Zhiqi Pang *et al.* (26) presentaron un método híbrido que combina técnicas supervisadas y no supervisadas. Este método utiliza una arquitectura TC-GAN para generar imágenes artificiales etiquetadas y transferir la persona de la imagen de entrada al fondo de la imagen del estilo deseado. Además, se propone el modelo de re-identificación DFE-Net, el cual emplea una versión modificada de la red ResNet-50 (27) preentrenada con la base de datos ImageNet (28), que tiene como entrada tanto imágenes reales sin etiquetar como generadas artificialmente. La red se utiliza como extractor de características de las imágenes para su posterior comparación. También en el año 2021, Yuanyuan Li *et al.* (29) propusieron el uso de una CycleGAN(11) y una red siamesa (16). En este caso, se utilizan datos etiquetados como dominio de entrada y se transfieren los estilos de imágenes de un dominio sin etiquetar. Por último, Xianjun Luo *et al.* (30) propone la arquitectura FFGAN. Para la generación de las imágenes artificiales utiliza la arquitectura CycleGAN (11). La innovación del trabajo se encuentra en el modelo de re-identificación el cuál es capaz de extraer las características locales, globales y semánticas de cada imagen para mejorar el rendimiento del modelo.

2.1. TRANSFERENCIA DE ESTILO ENTRE DISTINTOS DOMINIOS15

Año	Nombre	Modelo GAN	Transferencia	Base de datos	Github
2018	CamStyle (3)	CycleGAN	1 a 1	Misma/Distintas	Sí
2018	cmGAN (12)	Nuevo	1 a 1	Misma	No
2018	M2M-GAN (14)	Nuevo	Varios a Varios	Distintas	No
2018	SPGAN (15)	SiaNet CycleGAN	1 a 1	Distintas	No
2019	CTGAN (17)	StarGAN	1 a 1	Distintas	No
2019	DG-NET (7)	Nuevo	Combinaciones	Misma	No
2019	CTGAN (22)	Nuevo	Combinaciones	Misma	No
2020	CGAN-TM (19)	CycleGAN	1 a 1	Distintas	No
2020	UD-GAN (20)	Nuevo	1 a 1	Distintas	No
2020	cTransNet (21)	StarGAN	1 a Varios	Misma	No
2020	UnityGAN (23)	DiscoGAN/CycleGAN	1 a Genérico	Misma	No
2020	SECGAN (25)	CycleGAN	1 a 1	Misma	No
2021	TC-GAN (26)	Nuevo	1 a 1	Misma	No
2021	STrans (29)	Nuevo	1 a Genérico	Misma	No
2021	FFGAN (30)	CycleGAN	1 a 1	Misma	No

Tabla 2.1: Tabla con todas las arquitecturas de redes propuestas. (Año) fecha de presentación del artículo, (Nombre) nombre de la arquitectura propuesta, (Modelo GAN) si utilizan o se basan en alguna red generativa adversaria existente, (Transferencia) los dominios entre los que se transfieren los estilos y (Base de datos) si se ha globalizado el modelo, es decir, sí el entrenamiento se hecho con una base de datos y las pruebas con otra diferente.

2.2. Modificación de la postura

La re-identificación de personas presenta uno de los mayores desafíos debido a la gran variación en la postura de una persona en diferentes cámaras. Para abordar este problema, se propone generar nuevos datos de una misma persona mediante la modificación de su postura utilizando diversas arquitecturas. La generación de nuevos datos se basa en la extracción de la persona de la imagen original, que puede realizarse a través de la obtención de las articulaciones o mapas de calor (Fig. 2.9), y se adapta a la postura deseada para aumentar la cantidad y variabilidad de datos disponibles.

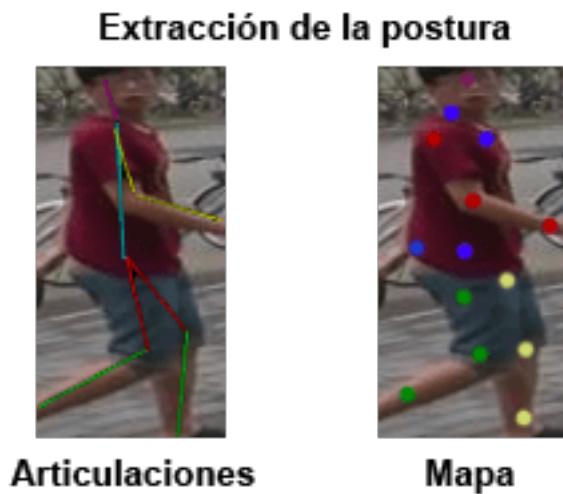


Figura 2.9: Tipos de extracción de la postura. Mediante articulaciones o mapas de calor.

En el año 2018, Xuelin Qian et al. (31) presentaron la arquitectura PN-GAN, capaz de generar imágenes artificiales de una persona en ocho posturas distintas. Las ocho posturas canónicas se obtuvieron mediante el algoritmo K-Means en la distribución de todas las imágenes de la base de datos. La herramienta Open Pose (32) fue utilizada para generar la plantilla, la cual detecta 18 articulaciones del cuerpo humano y sus uniones, y mediante el mapa de articulaciones de ambas imágenes, es posible transferir la postura de cada una de las ocho posturas canónicas a las imágenes de entrada, como se muestra en la Fig. 2.10.

En un trabajo similar, Aliaksandr Siarohin et al. (33) propusieron el De-

formable GAN, con el objetivo de generalizar el modelo anterior. Para ello, fue necesario entrenar el modelo de manera supervisada con pares de imágenes de la misma persona en diferentes posturas. Utilizaron también el método Open Pose (32) para obtener el mapa de articulaciones del cuerpo humano, descomponiendo en 18 articulaciones y un total de 10 subdivisiones, cabeza, torso, ambos brazos y piernas. Este modelo permitió transferir la postura de una persona en una imagen A a otra persona en una imagen B.

En otro trabajo, Yaoyu Li et al. (34) también emplearon Open Pose (32) como extractor de articulaciones y mapa de color con 19 ubicaciones en diferentes partes del cuerpo, lo que proporcionó robustez al modelo. El entrenamiento se realizó con pares de imágenes de una misma persona en diferentes posturas, de manera similar al trabajo anterior. La arquitectura permitió transferir la postura de una imagen a otra.

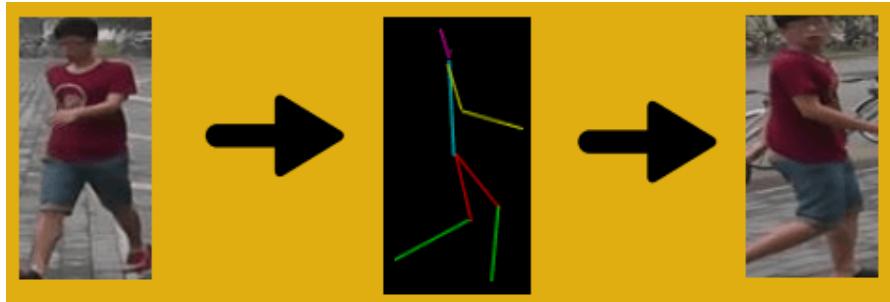


Figura 2.10: Transformación de la postura mediante una plantilla.

En el artículo del año 2018 de Yixiao Ge et al. (35), se propuso la arquitectura FD-GAN para transferir la postura de una imagen a otra. La red estuvo compuesta por un generador y dos discriminadores, uno para la identidad de la persona y otro para la postura, donde utilizaron la arquitectura PatchGAN (36). Utilizando una ResNet-50, extrajeron un vector de características de la imagen de entrada. Para la postura destino, utilizaron un mapa de 18 canales, donde cada uno representa la ubicación de un punto de referencia de la postura y lo convirtieron a un mapa de calor de tipo Gaussiano. Posteriormente, se convirtió a un vector de características de tamaño 128, y con estos dos datos, el modelo generó una nueva imagen de la misma persona con la postura especificada en el mapa de calor.

En el año 2019 Alessandro Borgia *et al.* (37) siguiendo la misma linea de las arquitecturas anteriores, extrayendo las articulaciones mediante el méto-

do Open Pose (32), proponen una arquitectura qué, en vez de evaluar una imagen en concreto, evaluará las secuencias de video del movimiento de una persona. Se predefinen ocho posturas canónicas las cuales serán tres mirando al frente, tres de espaldas, una de perfil mirando al lado derecho y otra de lado izquierdo. En primer lugar se obtienen las secuencias de video de una persona y se buscan las imágenes correspondientes con las posturas canónicas mediante la distancia euclíadiana, si no existe alguna de esas ocho posturas no existe, se generará la imagen artificial. Lo mismo ocurre con todas las personas que se encuentran en el video, se obtienen sus secuencias y si no existe alguna posición canónica la genera, mediante la distancia coseno comparan las ocho imágenes correspondientes con las ocho posturas canónicas de la persona de entrada con todas las demás secuencia de imágenes de personas. Se hace una clasificación y la que tenga una menor distancia coseno se le asigna como si fuera la misma persona en el modelo de re-identificación.

En 2020, Chengyuan Zhang y otros autores (38) presentaron el modelo PAC-GAN, compuesto por dos partes. En la primera, CPG-Net, se utiliza una GAN condicional para generar imágenes de una persona desde la cámara A y luego convertirlas al punto de vista de la cámara B. También se generan nuevas imágenes con posturas desde distintos puntos de vista de diferentes cámaras, aumentando así la cantidad de datos. El modelo se entrena con las articulaciones extraídas por Open Pose y la propia imagen. En la segunda parte, utilizan Cross-GAN (39), desarrollado por el mismo equipo, como modelo de re-identificación. En el trabajo de Y. Zhang *et al.* (40) proponen PGAN, en este caso, la postura se obtiene con un mapa de calor y, mediante dos imágenes de entrada, transfieren la postura de una a otra. Esta arquitectura mejora el rendimiento obtenido con la arquitectura FD-GAN (35) del año 2018.

En el año 2021, Ni Ziyang *et al.*(41) propusieron una nueva arquitectura capaz de corregir las imágenes para que las personas que aparecen en ellas estén centradas y rectas. Para ello, se entrenó la base de datos con imágenes en las que se indicaba la posición correcta de las personas. Asimismo, en ese mismo año, Hao Chen *et al.*(42) propusieron la arquitectura GLD de red generativa adversaria, que utiliza una malla tridimensional que representa distintas posturas y es capaz de generar una imagen nueva de la misma persona de entrada con la postura correspondiente.

Año	Nombre	Extractor postura	Núm. Posturas	Base de datos	Github
2018	PN-GAN (31) (3)	Open Pose	8	Misma	Sí
2018	Deformable GAN (33)	Open Pose	Otra imagen	Misma	Sí
2018	Yaoyu Li <i>et al.</i> (34)	Open Pose	Otra imagen	Misma	No
2018	FD-GAN (35)	Open Pose	Otra imagen	Misma	Sí
2019	Pose-aware Regulation (37)	Open Pose	8	Misma	No
2020	PAC-GAN (38)	Open Pose	Otras imágenes	Misma	No
2020	PGAN (40)	Open Pose	Otras imágenes	Misma	Sí
2021	Ni Ziyang <i>et al.</i> (41)	No	No	Misma	No
2021	GLD (42)	Malla 3D	No lo indica	Misma	Sí

Tabla 2.2: Resumen con todas las arquitecturas propuestas. (Año) fecha de presentación del artículo, (Nombre) nombre de la arquitectura propuesta, (Modelo GAN) si utilizan o se basan en alguna red generativa adversaria existente, (Transferencia) los dominios entre los que se transfieren los estilos y (Base de datos) si se ha globalizado el modelo, es decir, sí el entrenamiento se hecho con una base de datos y las pruebas con otra diferente.

2.3. Imágenes artificiales aleatorias

En este apartado aparecen los artículos que genera imágenes de personas de manera aleatoria, con diferentes posturas, iluminación, colores, fondos (ver Fig. 2.3). Mediante diferentes métodos se etiquetan para el entrenamiento del modelo de re-identificación. En primer lugar se van a detallar los artículos donde utilizan como base para la etiquetación de las imágenes artificiales el algoritmo de 1980 LSR (*Label Smooth Regularization*) y durante el año 2015 Christian Szegedy *et al.* (43) lo aplican por primera vez para un problema de clasificación.

En el año 2017 se presenta el primer trabajo donde entrena el modelo de re-identificación con imágenes generadas de manera aleatoria. Zhedong Zheng *et al.* (44) utilizan la red generativa adversaria propuesta en el año 2016, DCGAN (45), para generar datos aleatorios y los etiqueta con una técnica que denominan LSRO (ver Fig. 2.11), *label smoothing regularization for outliers*, la cual es una modificación del trabajo de Christian Szegedy *et al.* (43). Esta técnica asigna a la imagen artificial generada el mismo valor en todas las clases, es decir se distribuye uniformemente por todas las clases.

En el año 2019, Yan Huang *et al.* (46) presentaron MpRL, *Multi-pseudo Regularized Label* (ver Fig. 2.11). A diferencia del trabajo anterior, este método genera una etiqueta en función de la probabilidad de similitud con cada una de las clases de entrenamiento. Utilizan DCGAN (45) como red generativa adversaria. Por otro lado, en ese mismo año, Jean-Paul Ainam *et al.* (47) propusieron para el entrenamiento de la red generativa adversaria DCGAN, la agrupación de imágenes de la base de datos utilizando el algoritmo de clasificación K-Means. Además, proponen una nueva técnica de etiquetado, SLSR *Sparse Label Smoothing Regularization* (ver Fig. 2.11), la cual etiqueta a las imágenes artificiales con una distribución parcial basada en el grupo de donde fueron generadas utilizando el algoritmo K-Means. La técnica de etiquetado se utiliza para clasificar las imágenes de la base de datos en diferentes categorías, como el género, la edad o el tipo de ropa. La técnica de etiquetado LSRO (Label Smoothing Regularization Optimization) ha sido ampliamente utilizada en el campo de la re-identificación de personas, pero puede tener algunos problemas, como la creación de etiquetas ambiguas o incorrectas. La técnica SLSR propuesta en este trabajo trata de resolver estos problemas. En lugar de crear etiquetas precisas para cada imagen, SLSR etiqueta las imágenes con una distribución parcial. Esto significa que la etiqueta para cada imagen se basa en el grupo al que pertenece esa imagen,

en lugar de una etiqueta precisa. Esta técnica reduce la ambigüedad y la creación de etiquetas incorrectas. En relación al proceso de etiquetado LSR, se destaca el trabajo de Saleh Hussin *et al.*, 2021 (48), quienes proponen el empleo de la red generativa StyleGAN (8) para la creación de nuevos datos. Para ello, se entrena con alguno de los dataset más utilizados en el campo de la re-identificación de personas y, una vez generadas las nuevas imágenes, se aplica el método LSRO (ver Fig.2.11), propuesto por Zhedong Zheng *et al.*, 2017 (44), como se ha explicado anteriormente.

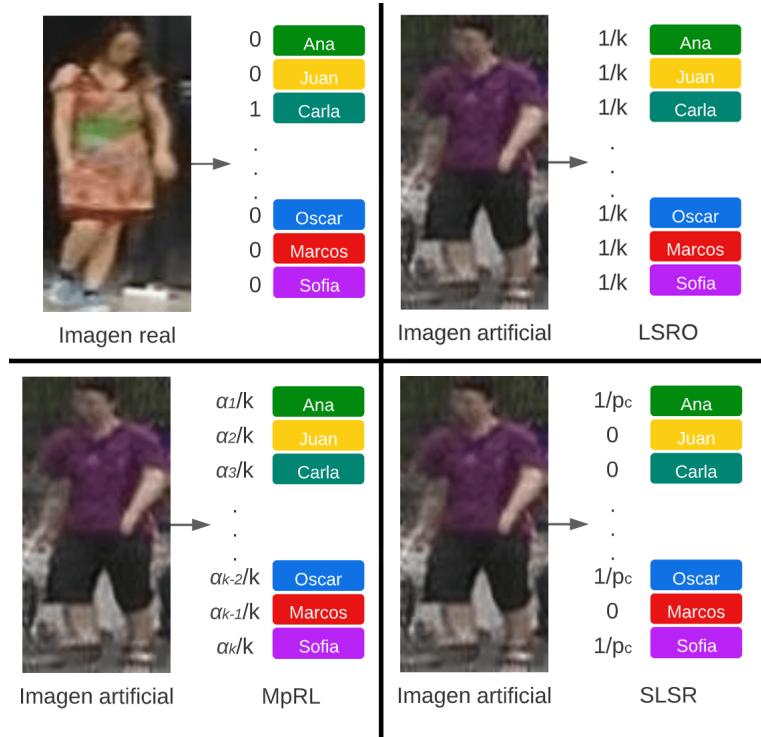


Figura 2.11: Diferentes tipos de etiquetados. Arriba a la izquierda etiquetando de una imagen real correspondiente a una persona. LSRO etiquetado de manera proporcional, k = número de clases, etiquetado MpRL en función de la similitud y SLSR etiquetado de manera proporcional en función del grupo al que pertenece, p_c = distribución en las clases del grupo al que pertenece.

En el año 2019 Chanho Eom *et al.*, 2021 (49) proponen una nueva arquitectura de red generativa adversaria, IS-GAN, *identity shuffle GAN* donde la generación de imágenes artificiales es a través de la interpolación entre

dos imágenes reales pudiendo discernir entre la parte superior y la inferior, las etiquetas de la imagen artificial serán las imágenes que han generado la interpolación.

Por último en el año 2021 Limin Xia *et al.*, 2021 (50) proponen una nueva arquitectura de red generativa adversaria, en primer lugar proponen el modelo MSSR (*Mixed-Space Super-Resolution model*) el cual mejora la resolución de las imágenes de entrada. Utilizan la arquitectura PGCN, *Part-based Graph Convolutional Net-work* para generar imágenes artificiales, y con esta misma red la utilizan para generar las etiquetas suaves de las imágenes artificiales (*soft multi-labels*).

Año	Trabajo	Modelo GAN	Etiquetado	Github	Github
2017	Zhedong Zheng <i>et al.</i> (44)	DCGAN	LSRO	Sí	Sí
2019	Yan Huang <i>et al.</i> (46)	DCGAN	MpRL	No	No
2019	Jean-Paul Ainam <i>et al.</i> (47)	DCGAN	SLSR	Sí	Sí
2019	Chanho Eom <i>et al.</i> (49)	IS-GAN	Imágenes	Sí	Sí
2021	Saleh Hussin <i>et al.</i> (48)	DCGAN	LSRO	No	No
2021	Limin Xia <i>et al.</i> , 2021 (50)	PGCN	<i>soft multi-labels</i>	No	No

Tabla 2.3: Tabla con todas las arquitecturas de redes propuestas. (Año) fecha de presentación del artículo, (Nombre) nombre de la arquitectura propuesta, (Modelo GAN) si utilizan o se basan en alguna red generativa adversaria existente, (Transferencia) los dominios entre los que se transfieren los estilos y (Base de datos) si se ha globalizado el modelo, es decir, sí el entrenamiento se hecho con una base de datos y las pruebas con otra diferente.

2.4. Revisiones del estado del arte

En los últimos años, ha habido un gran interés en el uso de redes generativas adversarias en modelos de re-identificación de personas. En 2019, Hamed Alqahtani *et al.* (51) proporcionaron una introducción detallada al estado del arte en este campo, describiendo diferentes tipos de redes generativas adversarias y 11 arquitecturas diferentes utilizadas para la transferencia de estilo, etiquetado LSRO y la globalización del modelo. Además, Zhiyuan Luo *et al.* (52) se centraron exclusivamente en arquitecturas que generan imágenes artificiales mediante un cambio de estilo entre distintas cámaras o bases de datos. Por último, Yiqi Jiang *et al.* (53) realizaron un estudio detallado sobre la calidad de las imágenes generadas por diferentes arquitecturas de redes generativas adversarias en modelos de re-identificación, analizando los detalles de estas imágenes artificiales que influyen en el rendimiento de los modelos de re-identificación. Concluyeron que no todas las imágenes generadas artificialmente son útiles para mejorar el rendimiento de los modelos de re-identificación.

Capítulo 3

Marco teórico

El marco teórico de esta investigación se basa en teorías y conceptos relacionados con la aumentación de datos de entrenamiento y re-identificación de personas. La aumentación de datos de entrenamiento se refiere a la técnica utilizada para aumentar el conjunto de datos de entrenamiento de un modelo de aprendizaje automático. Esta técnica puede mejorar el rendimiento de un modelo al agregar más datos de entrenamiento y permitir a un modelo aprender de manera más efectiva. En este contexto, StyleGAN es una red generativa adversaria que se utiliza para generar imágenes o videos sintéticos. Por otro lado en el campo de la re-identificación de personas, se han desarrollado diferentes modelos y técnicas para reconocer a las personas en imágenes y videos, los cuales suelen requerir un gran conjunto de datos de entrenamiento para poder funcionar de manera efectiva. Ambas disciplinas, redes generativas adversarias y modelos de re-identificación, son parte de una disciplina de inteligencia artificial denominada aprendizaje automático, la cual provee a una computadora la capacidad de aprender.

El aprendizaje de máquina es una disciplina que se fundamenta en la idea de que una máquina puede adquirir habilidades para desempeñar tareas complejas sin ser programada de manera específica para realizarlas. Se enfoca en el desarrollo de algoritmos y técnicas que permiten a una máquina aprender de manera automática a partir de datos. Es posible programar un ordenador para que aprenda de diversas maneras, tales como explorando la web, leyendo libros, jugando juegos o interactuando con personas. Un programa de aprendizaje de máquina puede aprender cualquier tarea que pueda ser codificada de forma matemática. Esta disciplina se emplea en una amplia variedad de aplicaciones, como el reconocimiento de patrones, la predicción

de resultados, la detección de fraudes y el procesamiento del lenguaje natural. Los algoritmos y técnicas de aprendizaje de máquina se utilizan en aplicaciones como sistemas de recomendación, asistentes personales, sistemas de diagnóstico médico y sistemas de detección de spam en correos electrónicos.

Dentro del aprendizaje máquina existe una rama que se denomina aprendizaje profundo. El aprendizaje profundo es una técnica de aprendizaje automático que se enfoca en el desarrollo de modelos de redes neuronales profundas. Las redes neuronales profundas son redes de computación que se inspiran en el funcionamiento del cerebro humano y se utilizan para realizar tareas complejas como el reconocimiento de patrones y la clasificación de imágenes. Se diferencia de otras técnicas de aprendizaje automático en que utiliza redes neuronales profundas con muchas capas de procesamiento. Estas capas de procesamiento permiten a las redes neuronales profundas aprender características complejas y abstractas en los datos de entrada y utilizarlas para realizar tareas complejas. Se utiliza en una amplia variedad de aplicaciones, como el reconocimiento de patrones, la clasificación de imágenes, el procesamiento del lenguaje natural y la generación de contenido sintético. Las redes neuronales profundas se utilizan en aplicaciones como sistemas de reconocimiento de voz, sistemas de recomendación y sistemas de diagnóstico médico.

3.1. Redes neuronales artificiales

Para comprender la importancia de las redes neuronales artificiales (RNA) es necesario conocer qué son estas redes. Una red neuronal artificial es un modelo simplificado del funcionamiento del cerebro, que está compuesto por unidades básicas de procesamiento llamadas neuronas. Estas neuronas se agrupan y se organizan en capas: la capa de entrada donde cada neurona artificial representa un dato de entrada, las capas ocultas y la capa de salida que extrae los datos de salida o destino. Las RNA son importantes porque permiten resolver problemas complejos en diferentes campos, como la medicina, la robótica, la informática y la industria, entre otros, gracias a su capacidad para aprender de manera autónoma y procesar grandes cantidades de información.

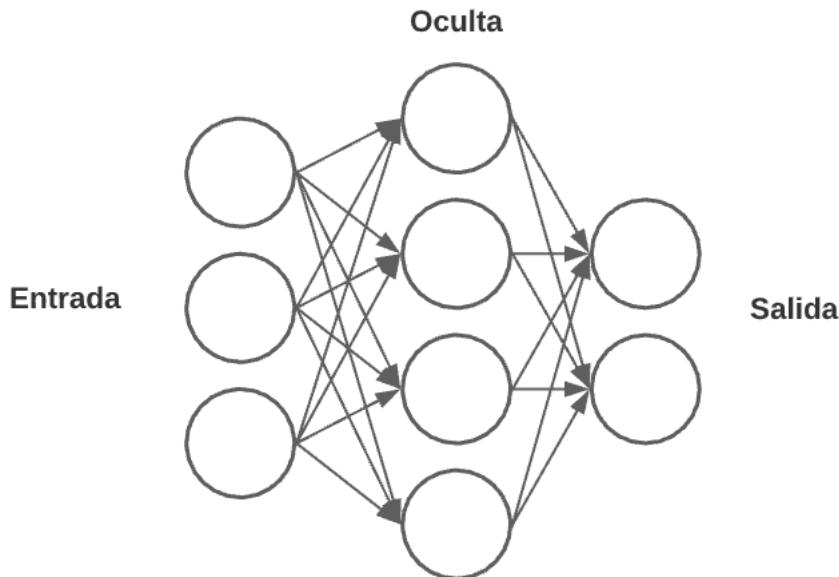


Figura 3.1: Ejemplo de red neuronal compuesta por una capa de entrada de tres neuronas, una capa oculta y una capa de salida de dos neuronas.

Las neuronas son similares a las neuronas biológicas, tienen conexiones de entrada a través de las cuales reciben estímulos, realizan un cálculo interno y generan un valor de salida.

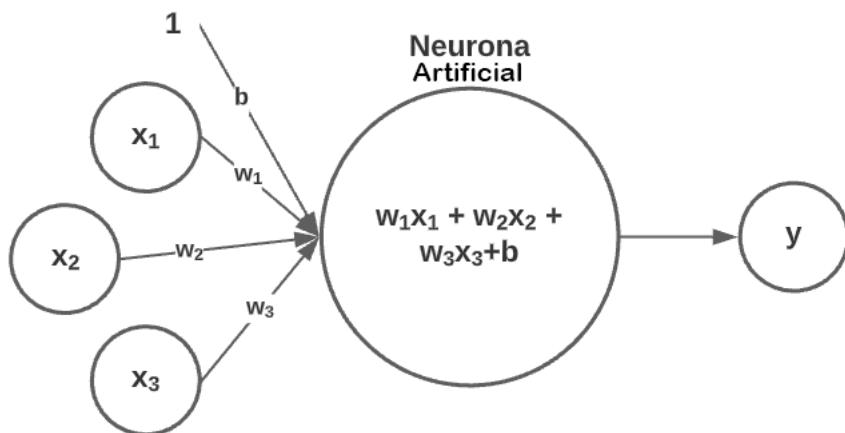


Figura 3.2: Ejemplo de una neurona compuesta por tres elementos de entrada x y sus respectivos pesos w .

La neurona artificial trabaja como una función, la cual realiza la suma ponderada de los valores de entrada por un valor llamado peso. Al multiplicarlos genera un valor que indica la intensidad con la que ese valor de entrada afecta a la neurona. La salida de la red neuronal artificial viene dada por la siguiente ecuación.

$$z = \sum_i^n x_i w_i + b \quad (3.1)$$

Donde y es la señal de salida de la neurona, f es la función de activación de la neurona, w_i son los pesos de los enlaces de la neurona, x_i son las señales de entrada de la neurona y b es el término independiente llamado sesgo, en inglés *bias*, que nos dará un mayor control en la función, básicamente es otra conexión con la neurona donde la variable de entrada tiene valor 1 y se pondrá controlar con el peso asignado a ella.

La función de activación de una neurona se utiliza para determinar la señal de salida de la neurona en función de la señal de entrada y de los pesos de sus enlaces. Las funciones de activación más comúnmente utilizadas en las redes neuronales artificiales son la función sigmoide, la función tangente hiperbólica y la función ReLU.

La función sigmoide es una función no lineal que toma un valor de entrada y lo transforma en un valor de salida en el rango de 0 a 1. Esta función se utiliza en las redes neuronales artificiales para modelar problemas de clasificación binaria, como la clasificación de imágenes en dos categorías. La función sigmoide se representa matemáticamente como:

$$f(x) = \frac{1}{1 + e^{-x}}$$

La función tangente hiperbólica es una función no lineal que toma un valor de entrada y lo transforma en un valor de salida en el rango de -1 a 1. Esta función se utiliza en las redes neuronales artificiales para modelar problemas de clasificación multiclase, como la clasificación de imágenes en múltiples categorías. La función tangente hiperbólica se representa matemáticamente como:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

La función ReLU es una función no lineal que toma un valor de entrada y lo transforma en un valor de salida en el rango de 0 a ∞ . Esta función se utiliza

en las redes neuronales artificiales para modelar problemas de regresión, como la predicción de un valor numérico a partir de un conjunto de características. La función ReLU se representa matemáticamente como:

$$f(x) = \max(0, x)$$

La finalidad de una red neuronal artificial o RNA es encontrar los valores de los pesos de manera tal que minimice la función de costo o error, la cual refleja si el modelo se está aproximando al resultado deseado.

Las funciones de costo que se utilizan en una red neuronal artificial se pueden representar matemáticamente mediante las siguientes ecuaciones:

Función de entropía cruzada:

$$J = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Donde J es la función de costo, N es el número de ejemplos del conjunto de datos de entrenamiento, y_i es la salida esperada para el ejemplo i , y \hat{y}_i es la salida obtenida por el modelo para el ejemplo i .

Función de error cuadrático medio:

$$J = \frac{1}{2N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Donde J es la función de costo, N es el número de ejemplos del conjunto de datos de entrenamiento, y_i es la salida esperada para el ejemplo i , y \hat{y}_i es la salida obtenida por el modelo para el ejemplo i .

Función de distorsión binaria:

$$J = \frac{1}{N} \sum_{i=1}^N (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

Donde J es la función de costo, N es el número de ejemplos del conjunto de datos de entrenamiento, y_i es la salida esperada para el ejemplo i , y \hat{y}_i es la salida obtenida por el modelo para el ejemplo i .

Función de divergencia KL:

$$J = \frac{1}{N} \sum_{i=1}^N y_i \log\left(\frac{y_i}{\hat{y}_i}\right) - y_i + \hat{y}_i$$

Donde J es la función de costo, N es el número de ejemplos del conjunto de datos de entrenamiento, y_i es la salida esperada para el ejemplo i , y \hat{y}_i es la salida obtenida por el modelo para el ejemplo i .

La combinación en capas de varias neuronas y la creación de varias capas hace posible la creación de modelos neuronales artificiales más complejos, la cantidad de capas determinan la profundidad del modelo y es lo que da nombre al aprendizaje profundo o deep learning.

3.2. Redes generativas adversarias

Dentro de las redes neuronales adversarias en el año 2014 surgió un modelo propuesto en el trabajo de Ian Goodfellow *et al.* (10) donde presenta una idea revolucionaria denominada red generativa adversaria (GAN, por sus siglas inglés).

Una red generativa adversaria (GAN) es un tipo de modelo de aprendizaje automático que se utiliza para generar contenido sintético de alta calidad. Una GAN está formada por dos componentes: una red generadora y una red discriminadora. La red generadora se encarga de generar contenido sintético, mientras que la red discriminadora se encarga de evaluar la calidad del contenido generado por la red generadora.

La red generadora es una red neuronal artificial que recibe un vector aleatorio como entrada y devuelve una imagen sintética como salida. Está formada por varias capas de nodos o neuronas que se conectan entre sí mediante enlaces o pesos y utilizan funciones de activación no lineales para procesar señales de entrada y generar señales de salida. La red generadora se entrena mediante un algoritmo de aprendizaje automático que le permite mejorar su desempeño en la tarea de generación de contenido sintético.

La red discriminadora es una red neuronal artificial que recibe una imagen como entrada y devuelve un valor real como salida. Está formada por varias capas de nodos o neuronas que se conectan entre sí mediante enlaces o pesos y utilizan funciones de activación no lineales para procesar señales de entrada y generar señales de salida. La red discriminadora se entrena mediante un algoritmo de aprendizaje automático que le permite mejorar su desempeño en la tarea de evaluación de la calidad de las imágenes.

Las dos redes se entrenan de manera simultánea y en una competencia iterativa. La red generadora trata de generar imágenes lo más realistas

possible para engañar a la red discriminadora, mientras que la red discriminadora trata de detectar las imágenes sintéticas generadas por la red generadora. Esta competencia entre las dos redes permite que ambas mejoren su desempeño en sus respectivas tareas.

La función de costo global de la red generativa adversaria se puede definir como la suma de las funciones de costo individuales de las redes generadora y discriminadora, y se puede representar matemáticamente como:

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

Donde p_{data} es la distribución de las imágenes reales del conjunto de datos de entrenamiento, p_z es la distribución de los vectores aleatorios que se utilizan como entrada de la red generadora, D es la red discriminadora, y G es la red generadora.

La función de costo global $V(D, G)$ se puede interpretar como la suma de dos términos: el primero es el error de clasificación de la red discriminadora cuando se le presentan imágenes reales del conjunto de datos de entrenamiento, y el segundo es el error de clasificación de la red discriminadora cuando se le presentan imágenes sintéticas generadas por la red generadora. Se minimiza mediante un algoritmo de optimización que permite que la red generadora mejore en su tarea de generación de contenido sintético, mientras que la red discriminadora mejore en su tarea de evaluación de la calidad de las imágenes.

En la Fig. 3.3 se muestra el esquema de esta red y se puede ver como el Generador nunca tiene acceso a la base de datos de entrenamiento, solo se apoya con los datos obtenidos a través del Discriminador.



Figura 3.3: Arquitectura GAN genérica.

Mayoritariamente el uso de esta arquitectura es para la generación de imágenes, aunque también sirve para cualquier tipo de datos como puede ser para la creación de audio o texto entre otros. Actualmente también se está utilizando para mejorar los gráficos, que parezcan realistas, de algunos video juegos, o la generación de videos en tiempo real.

3.2.1. StyleGAN

StyleGAN es una arquitectura de red generativa adversaria (GAN) desarrollada por NVIDIA en el año 2018 (54). Esta arquitectura ha sido entrenada para ser capaz de generar imágenes de caras de personas que no existen de alta calidad. En este caso fue entrenada con la base de datos FFHQ, la cual consta de imágenes de caras de personas de la red social Flickr. Utiliza una estructura de red generativa basada en capas de estilos que permite controlar de forma independiente distintos aspectos de la imagen generada, como la pose, la expresión facial, el género, etc.

La red generadora consta de un módulo de encoder y un módulo de generador, el módulo de encoder convierte la imagen de entrada en un tensor de estilos, que es un vector de dimensiones fijas que representa los distintos aspectos de la imagen. El tensor de estilos se utiliza como entrada del módulo de generador, que es una red generadora que utiliza una estructura de capas

de estilos para generar una imagen sintética que se aproxima a la imagen de entrada. Una vez entrenada, la red generadora puede utilizarse para generar imágenes sintéticas de alta calidad que se aproximan a las imágenes del conjunto de datos de entrenamiento. Además, la estructura de capas de estilos permite controlar de forma independiente distintos aspectos de las imágenes generadas, como la pose, la expresión facial, el género, etc. Esta capacidad de control en las imágenes generadas permite utilizar StyleGAN en aplicaciones como la aumentación de datos para modelos de re-identificación.

Una vez entrenada, la red generadora puede utilizarse para generar imágenes sintéticas de alta calidad que se aproximan a las imágenes del conjunto de datos de entrenamiento. El Generador es un espacio latente multidimensional y se muestra como un espacio matemático en el que se representan los vectores de entrada y salida de una red neuronal artificial. Este espacio se llama “*latente*” porque no se observa directamente, sino que se infiere a partir de las observaciones de entrada y salida. En este caso, está compuesto por 512 dimensiones, y a cada una de las posiciones corresponde con una imagen (ver Fig. 3.4), al introducir como entrada en el Generador un vector numérico aleatorio de tamaño 512, este nos generará una imagen de una cara artificial. Esto quiere decir que cada una de las imágenes que se pueden generar está compuesta por un vector latente de 512 posiciones, la Fig. 3.5 se muestra un ejemplo del espacio latente de un Generador de dos dimensiones.

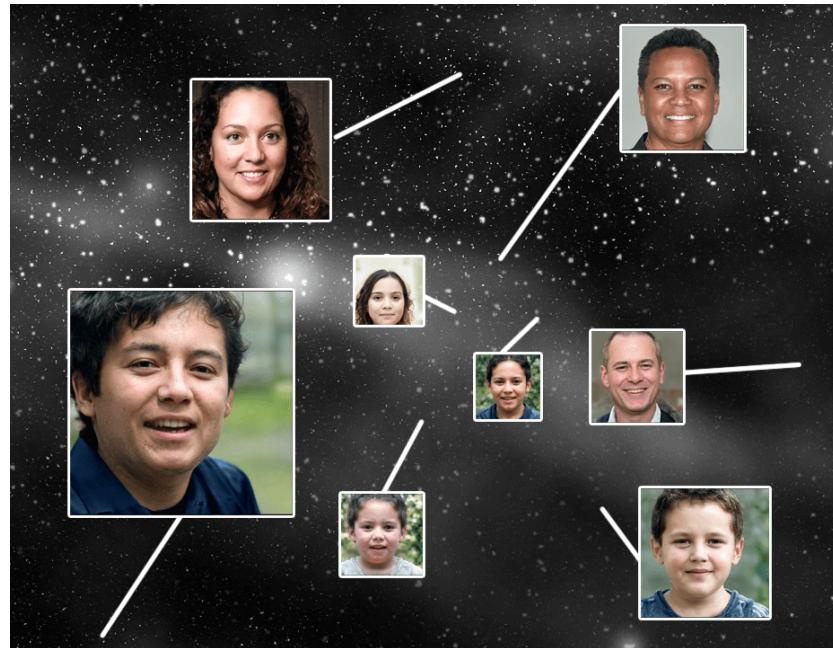


Figura 3.4: Representación gráfica de un Generador. Espacio multidimensional donde cada posición representa una imagen.

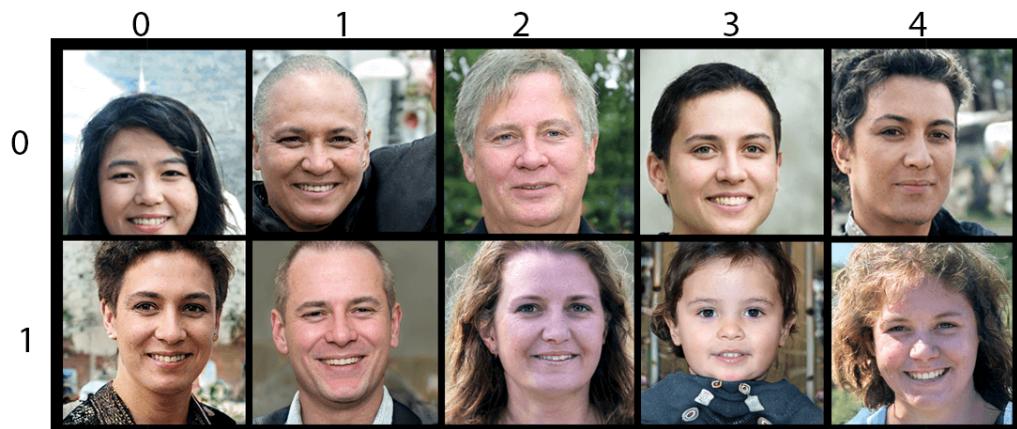


Figura 3.5: Ejemplo Generador de 2 dimensiones, a cada coordenada le pertenece una imagen. Al introducir el vector latente $[0,0]$ como salida se obtiene la imagen superior izquierda

Otra de las grandes propiedades de StyleGAN fue la capacidad de arreglar un error llamado enredo o en inglés “*disentangle*“ que tienen las redes generativas adversarias, se produce cuando las imágenes generadas por la red generadora se enredan entre sí, es decir, cuando se mezclan o confunden distintos aspectos de la imagen, como la pose, el género, la expresión facial, etc. Imaginemos que tenemos los vectores latentes de dos imágenes de caras, la primera es la cara de una niña y la segunda es de una mujer adulta, si hacemos la interpolación entre esas dos imágenes en un modelo que tiene el problema del enredo nos podríamos encontrar con imágenes totalmente aleatorias entre medias, mientras que utilizando el modelo StyleGAN la interpolación tiene coherencia y es muy suave. La Fig. 3.6 ejemplifica de manera gráfica este problema.



Figura 3.6: Problema del enredo. De la imagen 1 se hace una interpolación a la imagen 2. a) interpolación sin el problema del enredo. b) interpolación con el problema del enredo. Cómo se puede observar la interpolación a) es mucho más suave y coherente.

Su arquitectura está basada en las redes generativas progresivas o “*progressive gan*“, es un modelo de las redes generativas adversarias qué permite generar imágenes de mayor resolución. Durante la generación de la imagen comienza con imágenes muy pequeñas de, por ejemplo 4x4 píxeles, y las va escalando hasta llegar al tamaño de imagen deseado. Este método es muy eficaz para obtener imágenes de gran calidad. En la imagen[arquitectura] de la arquitectura de StyleGAN se puede ver como la imagen comienza con un tamaño de 4x4 píxeles y va en aumento hasta, en este caso, un tamaño de 1024x1024. Cada vez que se incrementa el tamaño de la imagen, pasa por una capa AdaIN o en inglés “*Adaptive Instance Normalization*“ .

AdaIN se basa en la idea de que el estilo de una imagen se puede representar mediante la distribución de sus características visuales. Por lo tanto, al transferir el estilo de una imagen a otra, se está modificando la distribución de sus características para que se asemeje a la distribución de la imagen fuente. Matemáticamente, esto se puede representar como una operación de normalización y escalado de las características visuales de la imagen de destino. Consideremos una imagen de destino x y una imagen fuente s . La operación de AdaIN se puede escribir como:

$$y = \frac{x - \mu_x}{\sigma_x} \odot \sigma_s + \mu_s$$

En esta ecuación, μ_x y σ_x representan la media y desviación estándar de las características visuales de la imagen de destino x , respectivamente. Por otro lado, μ_s y σ_s representan la media y desviación estándar de las características visuales de la imagen fuente s .

La operación \odot representa el producto elemento a elemento entre dos vectores, y se utiliza para aplicar la desviación estándar de la imagen fuente a la imagen de destino normalizada.

La operación de AdaIN se puede interpretar como una normalización de la imagen de destino, seguida de un escalado de sus características visuales utilizando la desviación estándar de la imagen fuente. De esta manera, se logra que la imagen de destino tenga el mismo estilo que la imagen fuente, mientras que se mantiene su contenido original.

Durante los experimentos el equipo de StyleGAN probó a modificar el vector latente en cada capa AdaIN, y como resultado la imagen mostraba diferentes cambios sobre la original. Por ejemplo, partiendo de un vector latente de la cara de una persona si se modifican los vectores latentes de las primeras capas de AdaIN notaron que cambiaban lo que ellos denominaron las características fuertes de la imagen, las siguientes representaban las características medias y por último las características suaves, en las Figs 3.7 3.8 3.9 se puede ver varios ejemplos reales modificando el vector latente en diferentes capas de AdaIN.

1. Características duras o en inglés “coarse”. Son las cuales se modifica la estructura de la imagen, en este caso la cara. Generando personas totalmente distintas.

2. Características medias o en inglés “*middle*”. La estructura de la imagen es igual a la original salvo que se modifica la postura
3. Características suaves o en inglés “*fine*”. Tanto la estructura y postura es la misma, se modifican cosas como el color de piel, pelo, ojos, etc.

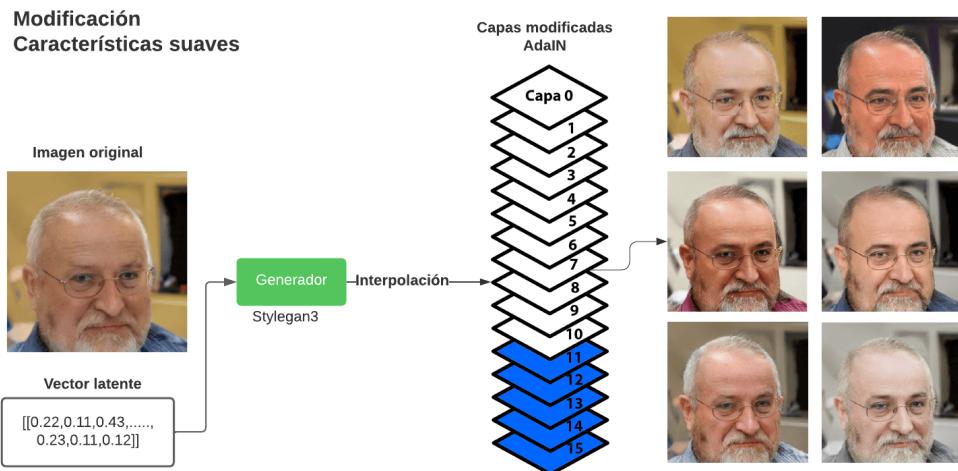


Figura 3.7: Se modifica el vector latente en las últimas capas de AdaIN . Se puede observar los cambios en las características suaves, como puede ser el color del pelo, ojos, piel.

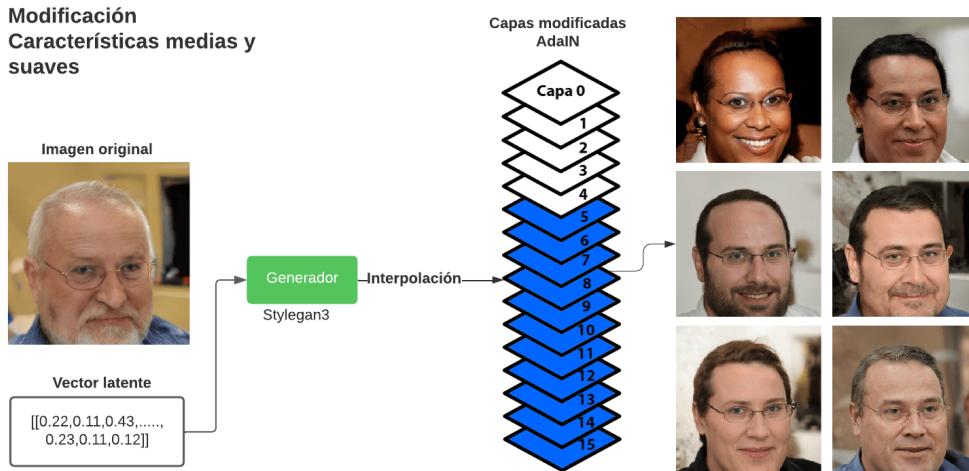


Figura 3.8: Se modifica el vector latente en las capas del medio y últimas de AdaIN ■. Se puede observar los cambios en las características medias y suaves. Se modifica la estructura de la cara mientras que la postura es la misma.

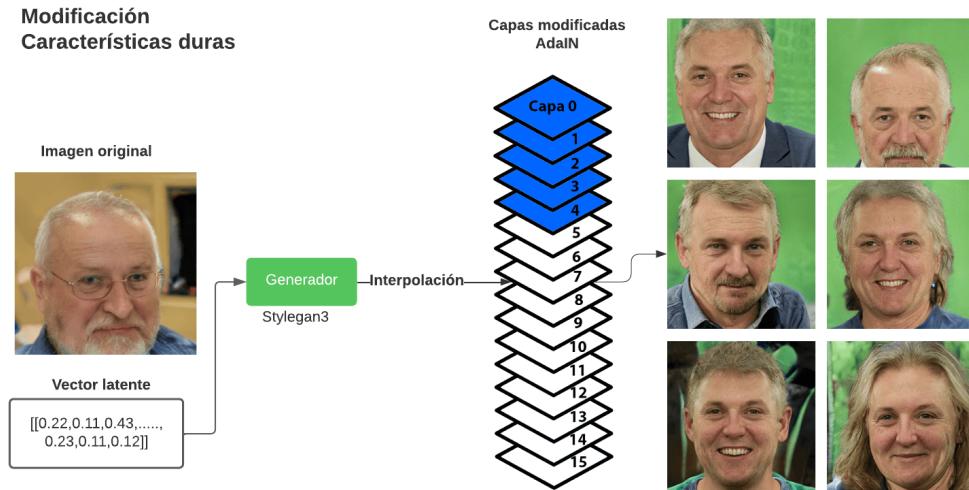


Figura 3.9: Se modifica el vector latente en las primeras capas de AdaIN ■. Se puede observar los cambios en las características fuertes, en la estructura de la imagen, mientras que el color de pelo o piel es el mismo.

La flexibilidad de StyleGAN la convierte en una arquitectura sumamente atractiva, no obstante, presenta una limitación importante. Como se puede apreciar en su arquitectura, las imágenes siempre son generadas a partir de un vector latente, mientras que otras arquitecturas de redes generativas adversarias cuentan con la capacidad de crear variaciones en imágenes preexistentes. Para conseguir el objetivo de alterar una imagen real con StyleGAN, se utilizan codificadores, conocidos en inglés como “*encoders*”. Un codificador es un componente de una red generativa adversaria (GAN) que se encarga de transformar una imagen de entrada en un vector de características latentes. Este vector de características latentes representa la imagen de entrada en un espacio latente, que se utiliza como entrada para la parte generativa de la GAN.

Codificador StyleGAN

Dentro de la literatura científica existen varios codificadores de StyleGAN. Su funcionamiento es bastante sencillo: una red neuronal toma una imagen como entrada y produce como salida los vectores latentes, los cuales se alimentan al Generador de StyleGAN para generar una imagen real inicial con la ayuda de StyleGAN. Independientemente de si estas imágenes se utilizaron o no en el entrenamiento del Generador, cada imagen de entrada siempre encontrará una imagen correspondiente dentro del espacio latente del Generador (Fig. 3.10).



Figura 3.10: Izquierda - imágenes reales. Derecha - obtenidas a través del codificador.

Explicación genérica de cómo funciona el entrenamiento de un codificador de StyleGAN.

1. Se introduca una imagen de la misma base de datos con la que se entrenó el Generador de StyleGAN.
2. El modelo a entrenar genera sus vectores latentes.
3. Introduce los vectores latentes en el generador preentrenado de StyleGAN y obtiene una imagen de salida perteneciente al espacio latente.
4. Compara la imagen de entrada y la imagen de salida con alguna métrica para comparar imágenes, como puede ser LPIPS, y actualiza los pesos del codificador.
5. El generador de StyleGAN está preentrenado y nunca modifica sus pesos.

Una vez entrenado el codificador funciona de la siguiente manera:

1. Se introduce como entrada la imagen real en el codificador.
2. Como salida el codificador genera los vectores latentes correspondientes con la imagen.
3. Introduce los vectores latentes en el generador de StyleGAN para obtener la representación de la imagen real dentro del espacio latente.
4. Se hacen interpolaciones en el generador de StyleGAN para obtener variaciones de la imagen de entrada.

3.3. Modelo de re-identificación

Dentro de la literatura, nos encontramos soluciones desde el año 1996, como es el trabajo de Q. Cai *et al.* (55), que intentan solventar el problema de la re-identificación. En la actualidad, el método más extendido es la utilización de una red neuronal como extractor de características.

El funcionamiento del modelo de re-identificación es muy sencillo.

1. Se necesita una base de datos con imágenes etiquetadas de personas obtenidas a través de diferentes cámaras de seguridad.
2. Se entrena el modelo de manera supervisada para que pueda clasificar un determinado número finito de personas del conjunto de entrenamiento. Por ejemplo utilizando la base de datos Market 1501, se entrena para que pueda clasificar 751 personas, podemos decir que tenemos un modelo que es capaz de detectar 751 personas.

Una vez entrenado el modelo y para poder utilizarlo con el conjunto de pruebas se siguen los siguientes pasos:

1. Como entrada se introduce en el modelo cada una de las imágenes y como salida cada una obtendrá su vector de tamaño 751, número de personas con las que se entrenó. Representando la proporción de cada una de las 751 personas que representa a la persona de entrada. En la imagen 234 se puede observar de una manera más sencilla.
2. Obtenemos el vector que representa a la imagen de la persona que queremos buscar y se mide la distancia con cada uno de los demás vectores que representan a las demás imágenes de personas. Dentro de la literatura es recurrente el uso de la distancia coseno o distancia euclíadiana, ambas son medidas de similitud entre dos vectores en un espacio vectorial.
3. Se hace una clasificación por distancia de todas las imágenes donde estamos buscando a esa persona y la que tenga menor distancia significa que la imagen es más similar a la original, significando que es probable que sea la misma persona.

No existe un modelo que te devuelva si es o no la misma persona. Se utiliza el modelo de clasificación como extractor de características y con base a ellas se calcula la distancia entre el vector de la imagen de la persona que se desea buscar y el resto de las imágenes.

Capítulo 4

Metodología

En este capítulo se aborda de manera detallada el enfoque utilizado para llevar a cabo el estudio. Se explican los métodos y técnicas empleados para la recolección y análisis de los datos, así como las herramientas y plataformas utilizadas. Asimismo, se proporcionan detalles técnicos sobre la generación de imágenes y el entrenamiento del modelo de re-identificación. La metodología propuesta para el desarrollo del trabajo se ha dividido en dos secciones. La primera sección se enfoca en el entrenamiento de la red generativa adversaria y la generación de imágenes artificiales. La segunda sección, por su parte, se centra en el entrenamiento y funcionamiento del modelo de re-identificación. La metodología se puede resumir con los siguientes puntos:

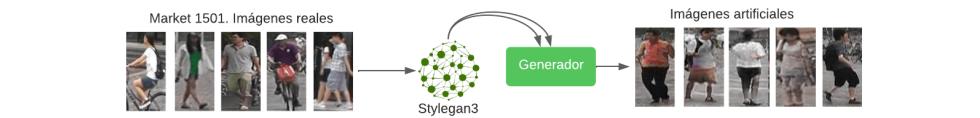
Generación de imágenes artificiales

- Entrenamiento de la red generativa adversaria StyleGAN3.
- Generación de multiples imágenes de personas artificiales en diferentes posturas.
- Utilizando como base una imagen de una persona real de la base de datos, generar imágenes artificiales de esa misma persona en diferentes posturas.
- Filtrado de imágenes. Eliminación automática de las imágenes generadas que presenten ruido o hayan sido generadas de manera incorrecta.

Modelo de re-identificación

- Diseño de la arquitectura y entrenamiento del modelo de re-identificación.
- Ejecución de pruebas del modelo de re-identificación.

a) Entrenamiento Stylegan3



b) Generación de imágenes artificiales

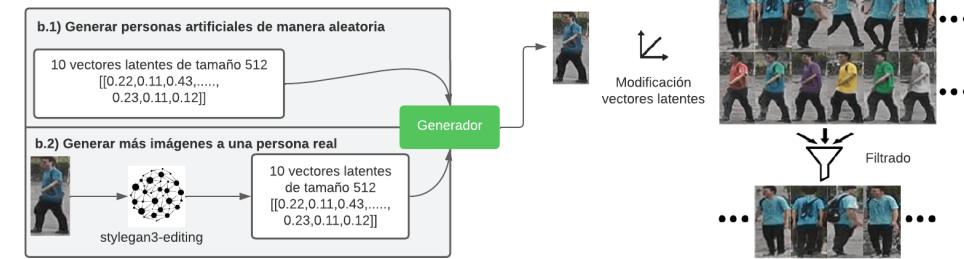
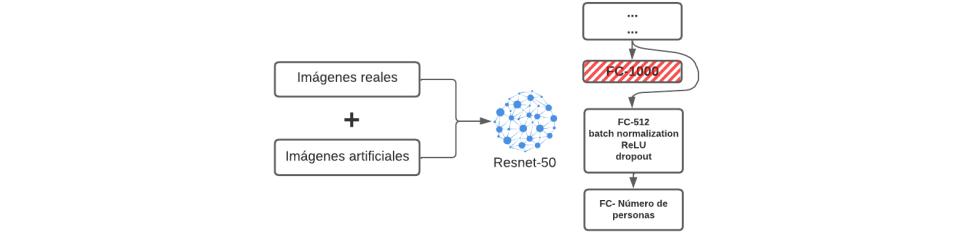


Figura 4.1: Arquitectura - Generación de imágenes artificiales.

c) Entrenamiento modelo re-identificación



d) Pruebas modelo re-identificación

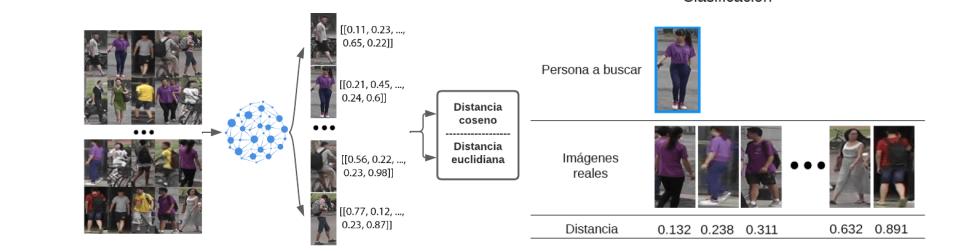


Figura 4.2: Arquitectura - Re-identificación.

4.1. Generación de imágenes artificiales

Para la generación de imágenes artificiales se utilizará la arquitectura de la red generativa adversaria StyleGAN3 (2). Es un modelo generativo de imágenes desarrollado por el equipo de investigación de Nvidia en el año 2021. Se trata de una versión mejorada del modelo StyleGAN2, que se caracteriza por su capacidad para generar imágenes de alta calidad y realismo en una amplia variedad de categorías de contenido.

StyleGAN3 utiliza un enfoque de aprendizaje profundo basado en generadores y discriminadores. El generador es una red neuronal que se entrena para generar imágenes que sean lo más realistas posible. Para ello, se le muestra un conjunto de imágenes reales y se le pide que genere imágenes que se asemejen a ellas. A medida que se entrena, el generador aprende a extraer características relevantes de las imágenes reales y a utilizarlas para generar imágenes que sean lo más realistas posible.

El discriminador es una red neuronal que se entrena para distinguir entre imágenes reales y generadas. Se le muestran tanto imágenes reales como generadas y se le pide que determine cuáles son reales y cuáles son generadas. A medida que se entrena, el discriminador aprende a identificar las características que diferencian a las imágenes reales de las generadas, y se utiliza para guiar el entrenamiento del generador hacia la generación de imágenes más realistas.

StyleGAN está pre-entrenada con 25 millones de imágenes de caras de las cuales 70 mil son reales de la base de datos FFHQ de alta calidad de resolución 1024×1024 píxeles y el resto fueron generadas por el Discriminador, su arquitectura más detallada en la Fig. 4.3. Actualmente, StyleGAN3 funciona como un generador de imágenes de caras de personas de gran calidad como se muestra en la Fig. 4.6. StyleGAN3 ha sido entrenado para generar caras y aplicaremos el proceso de aprendizaje transferido para reentrenarlo, como muestra la Fig. 4.4.

El aprendizaje transferido es una técnica en la que un modelo de aprendizaje automático que ha sido entrenado para realizar una tarea específica se utiliza como punto de partida para entrenar otro modelo para realizar una tarea diferente. En lugar de entrenar desde cero el nuevo modelo, se aprovecha el conocimiento y las habilidades adquiridas por el modelo original para iniciar el entrenamiento del nuevo modelo en un estado más avanzado. De esta manera, se reduce el tiempo y el esfuerzo necesarios para entrenar el nuevo

modelo y se mejora su rendimiento.

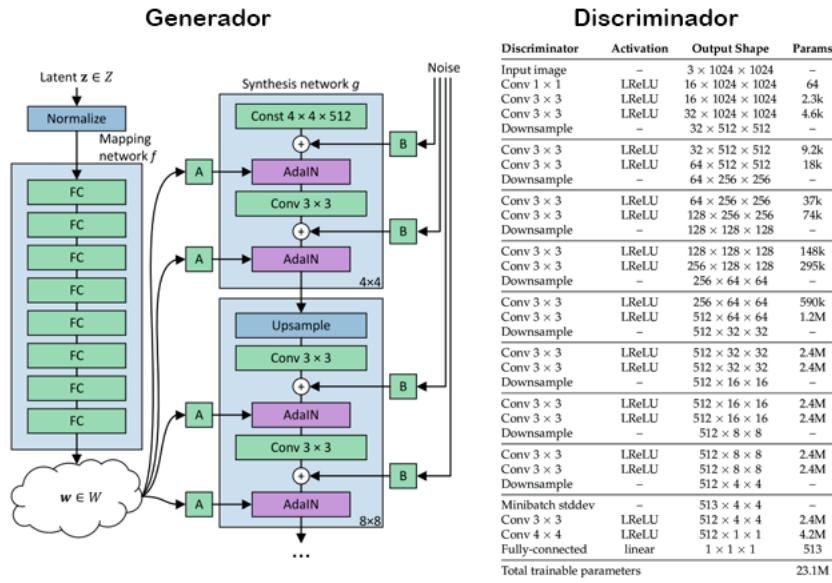


Figura 4.3: StyleGAN. Arquitectura del Generador y del Discriminador (8)

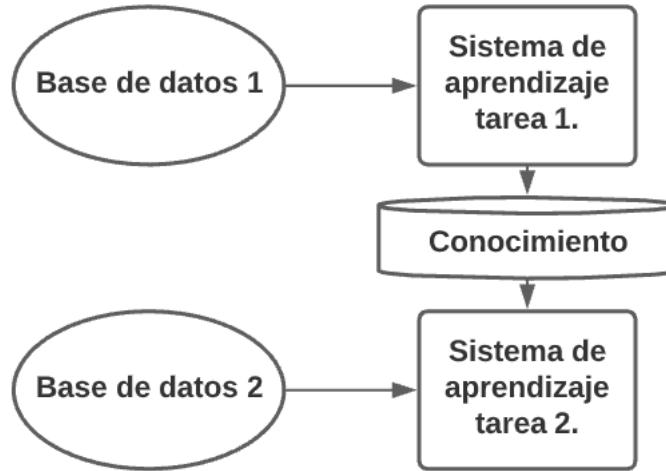


Figura 4.4: Diagrama - Transferencia de aprendizaje.



Figura 4.5: Personas generadas de manera artificial con StyleGAN3. Ninguna de estas personas existe.

Para el re-entrenamiento de StyleGAN3 se utilizó la base de datos Market-1501, la cual consta de 51,247 imágenes de 1,501 personas diferentes capturadas por seis cámaras distintas. Para evaluar el rendimiento de StyleGAN se utilizó la métrica "*Fréchet inception distance*" (FID), o en español, “*Distancia de Inicio de Fréchet*”, propuesta por P. Dimitrakopoulos *et al.*, 2017 (56). Esta métrica de distancia se utiliza para medir la similitud entre dos distribuciones de imágenes. La métrica FID se basa en la idea de que la distancia entre dos distribuciones de imágenes es la misma que la distancia entre las características de las imágenes extraídas de una red neuronal profunda. Para calcular la distancia FID entre dos distribuciones de imágenes, primero se extraen las características de cada distribución utilizando una red neuronal profunda, y luego se calcula la distancia entre esas características utilizando la distancia de Fréchet. Matemáticamente, la distancia FID entre dos distribuciones de imágenes se puede calcular de la siguiente manera:

$$\text{FID} = |\mu - \mu_w|^2 + \text{tr}(\sigma + \sigma_w - 2(\sigma\sigma_w)^{1/2}) \quad (4.1)$$

Compara la media y la covarianza de las imágenes reales y ficticias obteniendo los datos de una de las capas más profundas de la red neuronal, esta

capa está cerca a los datos de salida. Pretende imitar la percepción humana para identificar la similitud entre dos imágenes utilizando al Discriminador como un extracto de características. Si el valor obtenido es cero, indica que los datos generados y los datos reales son idénticos, lo cual significa que cuanto menor sea el valor obtenido, mayor será la similitud entre las imágenes generadas y las imágenes reales.



Figura 4.6: Imágenes de la base de datos Market-1501.

Se implementaron dos formas para generar imágenes, la primera es de manera totalmente aleatoria de personas artificiales, y la segunda es utilizando una imagen real de una persona y de ahí generar variaciones de la misma, Fig. 4.1.

- Generación de imágenes de personas artificiales.

Por medio de un número aleatorio, también conocido como semilla, el Generador asigna un vector latente que se corresponde con una imagen. Para obtener variaciones de la imagen original, se puede utilizar otro vector latente aleatorio, a través de otra semilla o mediante interpolaciones. En las diferentes capas AdaIN del modelo, también llamado mezcla de estilos, se va modificando el vector latente y se consiguen las diferentes variaciones de la imagen original. Es posible lograr desde un cambio total en la estructura de la imagen hasta cambios más sutiles, como el cambio de la tonalidad, iluminación, colores, saturación, entre otros. Otra forma de generar variaciones de la imagen original es a través de la modificación del vector latente original mediante interpolaciones, tal como se muestra en la Fig. 4.8.

En la Fig. 4.7 se puede ejemplificarse como se mezclan los vectores latentes para obtener variaciones de la imagen original.

- Generación de imágenes artificiales de personas reales.

Como se ha explicado anteriormente para eso se necesita entrenar un modelo, un codificador, que sea capaz de obtener el vector latente de una imagen real. En este caso se ha utilizado el modelo StyleGAN3-editing (57) y se ha entrenado sobre el codificador PsP desarrollado por Richardson *et al* (58). El codificador es parte de una red neuronal que se encarga de procesar la información de entrada y convertirla en una representación interna que pueda ser utilizada por StyleGAN3 para generar la versión de la imagen real dentro del espacio latente.

Para la función de pérdida se utiliza la métrica LPIPS en inglés *Learned Perceptual Image Patch Similarity* que va a comparar la imagen real con la obtenida en el generador. Es una medida de la distancia o diferencia entre dos imágenes en términos de su similitud percibida por un observador humano. Esta métrica se basa en una red neuronal preentrenada llamada VGG-16, que ha sido diseñada para reconocer patrones en imágenes. La idea detrás de LPIPS es que si dos imágenes tienen una distancia LPIPS pequeña, entonces se perciben como similares por un observador humano.

Matemáticamente, la métrica LPIPS se calcula de la siguiente manera: Primero, se utiliza la red VGG-16 para extraer una representación de cada una de las dos imágenes en cuestión. Esta representación se llama "*mapa de características*" y es un tensor de tres dimensiones que contiene información sobre las características visuales presentes en cada imagen. Denotamos estos mapas de características como f_1 y f_2 .

A continuación, se calcula la distancia euclídea entre los dos mapas de características. Esta distancia se interpreta como la similitud percibida entre las dos imágenes. Cuanto más pequeña sea esta distancia, más similares serán las imágenes. La distancia euclídea entre f_1 y f_2 se define como:

$$\text{dist}(f_1, f_2) = \|f_1 - f_2\|_2$$

En la Fig. 4.9 muestra el proceso de entrenamiento del codificador.

Una vez se obtiene el vector latente de la imagen real dentro del espacio latente de StyleGAN3 se procede a generar nuevas imágenes de la misma manera que en el caso anterior.

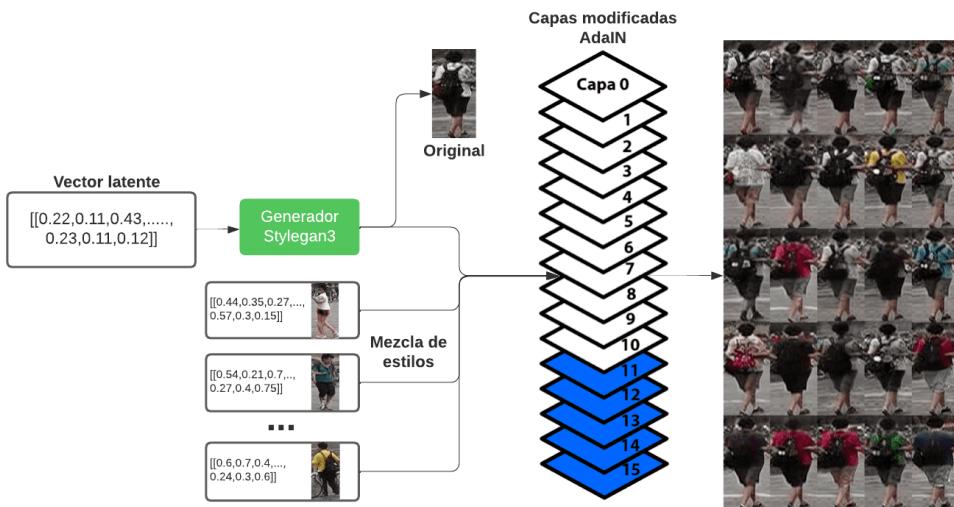


Figura 4.7: Partiendo del vector latente inicial se generan nuevas imágenes. Para mezclar los estilos en las capas . se le introduce el vector latente de las otras imágenes mientras que las capas blancas se le introduce el vector latente de la imagen original.

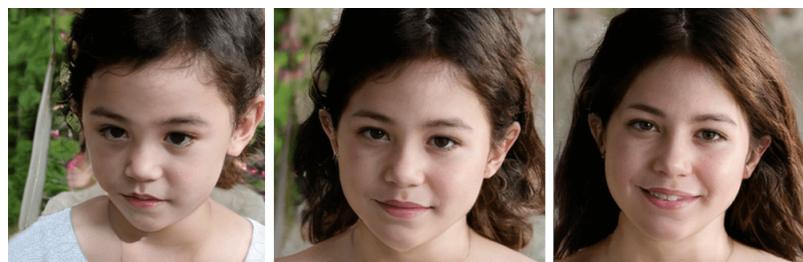


Figura 4.8: Ejemplo de interpolación desde una imagen de entrada hasta la misma image envejecida.

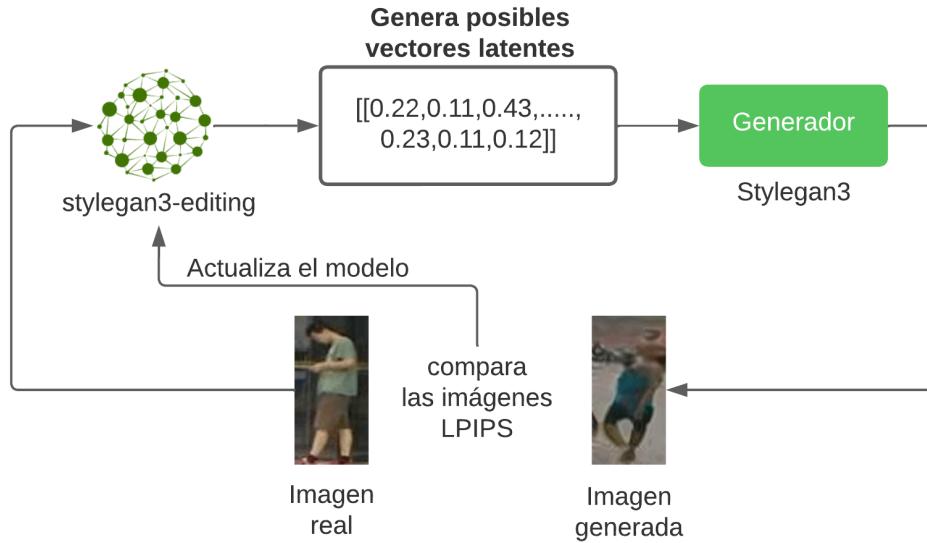


Figura 4.9: Entrenamiento del codificador de StyleGAN3.

Una vez generadas las imágenes se necesita automatizar el filtrado de las mismas. Puede ocurrir que se hayan generado imágenes con ruido o distorsionadas.

Para medir y descartar las imágenes generadas por la red generativa adversaria, se han usado métricas basadas en la calidad de los datos generados obtenidos. El primer filtro que se aplica es un modelo para la detección de peatones y para medir la similitud de las imágenes generadas se ha utilizado la métrica, *structural similarity index measure* (SSIM).

- **Filtrado YoloV4 tiny**

YOLOv4 tiny, un trabajo propuesto por Z. Jiang *et al* (59), consiste en una versión reducida del modelo YOLOv4, cuyo propósito es detectar objetos en imágenes y videos. YOLO (You Only Look Once) es una técnica de detección de objetos que se destaca por su rapidez y precisión. La versión “*tiny*” de YOLOv4 resulta particularmente útil para dispositivos de baja potencia, debido a que es menos exigente en cuanto a recursos y puede ejecutarse de manera eficiente en dispositivos móviles y computadoras de bajo rendimiento. En términos generales,

YOLOv4 tiny utiliza una red neuronal convolutional para extraer características de una imagen y luego utiliza una combinación de técnicas de aprendizaje automático para realizar la detección de objetos. La versión tiny de YOLOv4 se ha optimizado para detectar peatones con una precisión y velocidad comparables a las de otros modelos más grandes, pero con una menor carga en términos de recursos. Esto lo convierte en una excelente opción para aplicaciones en tiempo real en dispositivos con capacidades limitadas.

■ Filtrado SSIM

La métrica Structural SIMilarity (SSIM) es una medida de similitud estructural entre dos imágenes. La métrica SSIM se utiliza a menudo para evaluar la calidad de una imagen procesada en comparación con una imagen original, y se calcula comparando las características estructurales de ambas imágenes. SSIM se basa en el hecho de que la percepción humana de la calidad de una imagen se basa en su contenido estructural, y no solo en la diferencia de píxeles entre dos imágenes. Por lo tanto, la SSIM se utiliza para medir la similitud estructural entre dos imágenes y dar una puntuación que refleje la calidad percibida por un observador humano.

Para llevar a cabo el cálculo de la métrica SSIM, se comparan tres características estructurales de dos imágenes: su intensidad media, su varianza de intensidad y su covarianza de intensidad. La SSIM se obtiene a partir del producto de estas tres características, y se considera que dos imágenes poseen una alta SSIM si presentan una intensidad media similar, una varianza de intensidad similar y una covarianza de intensidad similar. El resultado se obtiene a partir del producto de estas tres características, y se denota como:

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad (4.2)$$

Donde x e y son las dos imágenes que se están comparando, $l(x, y)$ es la similitud en la intensidad media, $c(x, y)$ es la similitud en la covarianza de intensidad y $s(x, y)$ es la similitud en la varianza de intensidad.

La similitud en la intensidad media se calcula como:

$$l(x, y) = \frac{2 \cdot \mu_x \cdot \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (4.3)$$

La fórmula se lee como 'La luminosidad l entre dos imágenes x e y es igual al cociente entre el producto de dos veces la media μ de la imagen x y la media μ de la imagen y más el valor constante $C1$, y el cuadrado de la media μ de la imagen x más el cuadrado de la media μ de la imagen y más el valor constante $C1$.'

Donde μ_x y μ_y son las intensidades medias de las imágenes x e y , respectivamente, y $C1$ es una constante que se utiliza para evitar divisiones por cero.

La similitud en la covarianza de intensidad se calcula como:

$$c(x, y) = \frac{2 \cdot \sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (4.4)$$

La fórmula se lee como 'El contraste c entre dos imágenes x e y es igual al cociente entre el producto de dos veces la desviación estándar correlacionada σ de las imágenes x e y más el valor constante $C2$, y el cuadrado de la desviación estándar σ de la imagen x más el cuadrado de la desviación estándar σ de la imagen y más el valor constante $C2$ '.

Donde σ_{xy} es la covarianza de intensidad entre las imágenes x e y , σ_x y σ_y son las varianzas de intensidad de las imágenes x e y , respectivamente, y $C2$ es una constante que se utiliza para evitar divisiones por cero.

La similitud en la varianza de intensidad se calcula como:

$$s(x, y) = \frac{2 \cdot \sigma_x \cdot \sigma_y + C_3}{\sigma_x^2 + \sigma_y^2 + C_3} \quad (4.5)$$

La fórmula se lee como 'La similitud de bordes s entre dos imágenes x e y es igual al cociente entre el producto de dos veces la desviación estándar σ de la imagen x y la desviación estándar σ de la imagen y más el valor constante $C3$, y el cuadrado de la desviación estándar σ de la imagen x más el cuadrado de la desviación estándar σ de la imagen y más el valor constante $C3$ '.

Donde σ_{xy} es la covarianza de intensidad entre las imágenes x e y , σ_x y σ_y son las varianzas de intensidad de las imágenes x e y , respectivamente, y $C3$ es una constante que se utiliza para evitar divisiones por cero.

En resumen, la métrica SSIM se calcula comparando las intensidades medias, las covarianzas de intensidad y las varianzas de intensidad de dos imágenes. La SSIM se obtiene como el producto de la similitud en cada una de estas características, y se utiliza para evaluar la calidad de una imagen procesada en comparación con una imagen original.

La métrica SSIM se utiliza para evaluar la calidad de una imagen procesada en comparación con una imagen original, y se basa en la comparación de las características estructurales de ambas imágenes. Cuanto mayor sea el resultado más variación habrá en las imágenes generadas.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.6)$$

4.2. Modelo de re-identificación

Un modelo de re-identificación es un algoritmo utilizado en el procesamiento de imágenes y en la inteligencia artificial que permite identificar y rastrear objetos o personas en una secuencia de imágenes. Estos modelos se basan en la comparación de características visuales entre diferentes imágenes para determinar si se trata del mismo objeto o persona. Matemáticamente, un modelo de re-identificación utiliza una función de similitud para calcular la similitud entre dos imágenes. Esta función toma dos vectores de características visuales (uno de la imagen de referencia y otro de la imagen a comparar) y devuelve un valor que indica la similitud entre ambas imágenes. Si el valor devuelto por la función de similitud supera cierto umbral, se determina que las imágenes corresponden al mismo objeto o persona. Para calcular los vectores de características visuales, el modelo utiliza una red neuronal que ha sido entrenada previamente con un conjunto de datos de imágenes etiquetadas. La red neuronal extrae características relevantes de las imágenes y las agrupa en un vector de características. Estos vectores se utilizan luego en la función de similitud para determinar la similitud entre las imágenes.

La arquitectura propuesta en este trabajo se muestra en la Fig. 4.2. Se utiliza una red neuronal convolucional Resnet50, en la que se modifica la última capa para que la salida se adapte al número de personas con las que se entrenará el modelo. Durante el entrenamiento, se emplea como función de pérdida la entropía cruzada, también conocida como “*cross-entropy loss*” en inglés. Esta función se define como:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i$$

En esta ecuación, se tienen los siguientes términos:

- y : representa la etiqueta real o valor deseado de la salida.
- \hat{y} : representa la salida predicha por el modelo.
- N : es el número de ejemplos en el conjunto de datos.

La entropía cruzada es una función de pérdida comúnmente utilizada en problemas de clasificación, donde la salida del modelo se interpreta como la probabilidad de que un ejemplo pertenezca a cada clase. La idea detrás de la entropía cruzada es que, si la salida del modelo es una buena aproximación de la verdadera distribución de probabilidad, entonces la función de pérdida de entropía cruzada tendrá un valor bajo. En contraposición, si la salida del modelo es muy diferente de la verdadera distribución de probabilidad, entonces la función de pérdida de entropía cruzada tendrá un valor alto. Se utiliza para medir qué tan bien el modelo está haciendo predicciones sobre la distribución de probabilidad real de las clases. Durante el entrenamiento del modelo, se optimiza la función de pérdida para mejorar su precisión en las predicciones. Una vez finalizado el entrenamiento, el modelo funciona como un extractor de características y se procede a la clasificación de las imágenes. En este trabajo, se utilizó una red neuronal convolucional Resnet50, en la que se modificó la última capa para que la salida esté en función del número de personas con las que se entrena el modelo. Durante el entrenamiento, se utilizó la función de pérdida de entropía cruzada para optimizar la precisión del modelo.

Introducimos una a una todas las imágenes a evaluar dentro del modelo para obtener sus respectivos vectores de características y para clasificar qué

imágenes son de la misma persona se comparan cada uno de los vectores de las imágenes con el vector de la imagen original mediante la distancia coseno. Es una medida de similitud entre dos vectores en un espacio vectorial. Esta medida se calcula utilizando el coseno del ángulo entre los dos vectores y se puede interpretar como la proyección del vector más corto sobre el vector más largo.

Matemáticamente, la distancia coseno entre dos vectores \mathbf{a} y \mathbf{b} se puede calcular de la siguiente manera:

$$d_c(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}$$

En esta ecuación, $\mathbf{a} \cdot \mathbf{b}$ es el producto escalar de los vectores \mathbf{a} y \mathbf{b} , y $|\mathbf{a}|$ y $|\mathbf{b}|$ son las normas de los vectores \mathbf{a} y \mathbf{b} , respectivamente.

La distancia coseno tiene un valor entre 0 y 1, donde un valor más cercano a 1 indica una mayor similitud entre los vectores \mathbf{a} y \mathbf{b} , y un valor más cercano a 0 indica una menor similitud entre ellos. Despues de haber obtenido la distancia coseno de todas las imágenes se ordenan y las que tengan una menor distancia coseno serán las que más se acerquen a la imagen original, es decir serán las que haya detectado como imágenes de la misma persona. En la Fig. 4.2 en el apartado clasificación se muestra como se clasifican las imágenes mediante la distancia coseno.

Capítulo 5

Resultados experimentales

En este capítulo se muestran los resultados obtenidos durante la experimentación. Debido a la complejidad de la arquitectura se ha procedido a dividirla en dos secciones.

- Generación de imágenes

Resultados del entrenamiento, generación de imágenes artificiales y filtrado.

- Pruebas con el modelo de re-identificación

Los entrenamientos se han dividido en dos. En primer lugar se ha entrenado el algoritmo de re-identificación con diferente número de personas artificiales y, por otro lado, con diferente número de imágenes artificiales de personas reales de la base de datos.

5.1. Generación de imágenes

Se ha utilizado la red generativa adversaria StyleGAN3 (2) para la generación de imágenes artificiales. El modelo está pre-entrenado con 25 millones de imágenes de caras de las cuales 70 mil son reales de la base de datos Flickr-Faces-HQ Dataset (FFHQ) de alta calidad de resolución 512×512 píxeles y el resto fueron generadas por el discriminador.

En la Tabla 5.3 se presentan las características del entrenamiento de StyleGAN3, el cual consistió en una transferencia de aprendizaje y posterior re-entrenamiento con 51,247 imágenes provenientes de la base de datos Market-1501. En la Tabla 5.1 se detallan los hiperparámetros utilizados. Para medir

cfg	gpus	batch	gamma	kimg	snap	metrics
stylegan3-r	1	16	2	5000	20	fid50k_full

Tabla 5.1: Hiperparámetros utilizados durante el entrenamiento de StyleGAN3, “**cfg - stylegan3-r**” parámetro utilizado para determinar el tipo de entrenamiento “**config R**” o equivalente de rotación, hace unas pequeñas modificaciones en la red para que permita la rotación y traslación de las imágenes generadas, esto produce que no empeore la medición de la métrica FID si rotas o mueves las imágenes generadas. Se entrenó en una GPU. “**batch - 16**”, número de imágenes que se introducen en la red en cada iteración del entrenamiento. “**gamma - 2**”, Peso de regularización R1, la cual indica qué tan rápido se actualizan los pesos. “**kimg - 5000**”, duración total del entrenamiento. “**snap- 20**”, cada cuanto se guarda el modelo, en este caso cada 80,000 imágenes. “**metrics - fid50k_full**”, métrica utilizada para medir el rendimiento del modelo durante el entrenamiento.

Modelo	Img. entrenamiento	Img. pruebas	Entrenamiento	Hardware
StyleGAN3	51247	No aplica	2d 08h 24m	Titan RTX

Tabla 5.2: Datos técnicos del entrenamiento StyleGAN3, para la generación de imágenes artificiales. Modelo, número de imágenes utilizadas para el entrenamiento, duración del entrenamiento y la tarjeta gráfica utilizada.

el rendimiento del entrenamiento, se empleó la métrica Fréchet Inception Distance (60) (FID), la cual se aplicó tanto a las imágenes generadas como a las reales. Cuanto más cercano sea el valor de ambas, mejor habrá sido la generación de imágenes. Como se puede observar en la Tabla 5.4, el rendimiento de StyleGAN3 es muy superior al de otras redes generativas adversarias entrenadas con la misma base de datos. Como ejemplo de las imágenes generadas, se puede apreciar en la Fig. 5.2 la capacidad y calidad que tiene el modelo StyleGAN3 para generar imágenes artificiales en comparación con las reales.

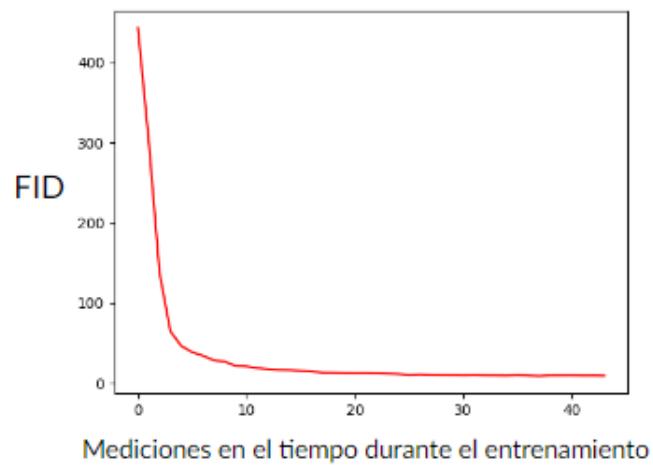


Figura 5.1: Evolución del rendimiento del modelo mediante la métrica FID en diferentes épocas del entrenamiento de StyleGAN3.

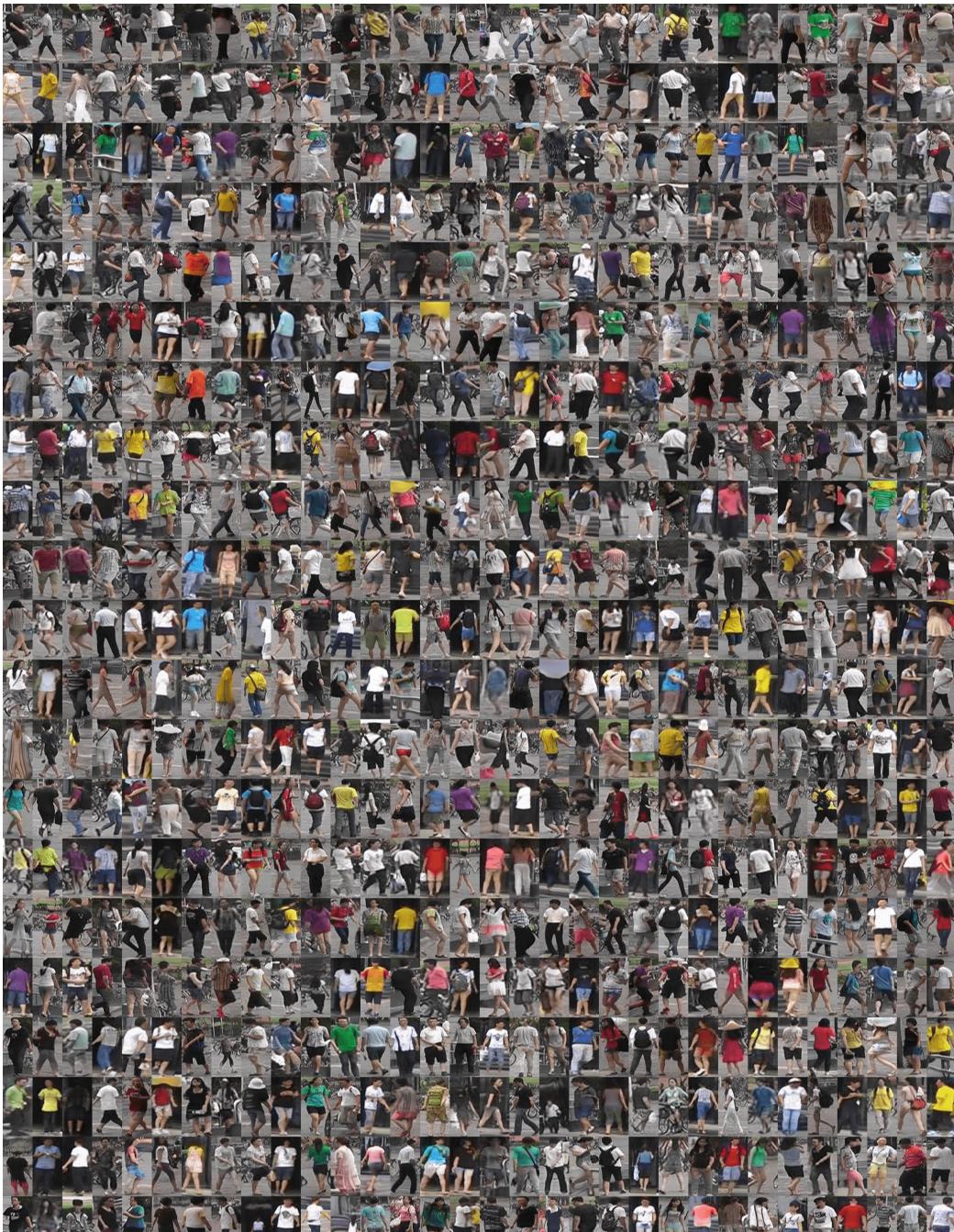


Figura 5.2: Imágenes artificiales generadas después del entrenamiento de manera aleatoria.

Para generar las variaciones de las imágenes de una misma persona se utilizó la mezcla de estilos. En el caso del modelo entrenado con Market-1501

Características	Capas AdaIN	Ejemplo
Suaves	(12,13,14,15)	
Medias	(5,6,7,8,9,10,11)	
Duras	(0,1,2,3,4,5)	

Tabla 5.3: Capas utilizadas para generar las nuevas imágenes de una misma persona en función de la modificación de sus características suaves, medias o duras.

Una vez generadas las imágenes artificiales se aplicaron dos filtros para descartar las imágenes que se hayan podido generar de manera incorrecta o contengan ruido.

- **Filtrado YoloV4 tiny**

Se utilizó el modelo entrenado YoloV4 tiny (59) para la detección de peatones en las imágenes generadas. Se descartaron todas las imágenes cuya clasificación por debajo del umbral de 0.6, el cual se determinó mediante el análisis de la Fig. 5.3, donde se muestran los diferentes porcentajes de imágenes clasificadas como no peatones utilizando distintos valores de umbrales sobre las imágenes reales de la base de datos Market-1501. Analizando estos datos, se observa que cuando se utiliza un umbral de 0.6, el porcentaje de imágenes mal clasificadas como no peatones es sólo del 6.45 %, considerándolo como un valor conservador para utilizar en el filtrado de las imágenes generadas artificialmente.

- Filtrado SSIM

Se utilizó la métrica de similitud estructural (SSIM, por sus siglas en inglés) (9) para evaluar la similitud entre dos imágenes. La metodología empleada para aplicar esta métrica consistió en seleccionar una imagen de una persona y compararla con el resto de las imágenes de esa misma persona en diferentes posturas. Si el valor de similitud era igual a uno, se consideraba que se trataba de la misma imagen. Esta métrica se aplicó a las imágenes reales de la base de datos Market-1501 y se obtuvo un histograma (ver Fig.5.4). A través de este histograma, se determinó que se descartarían aquellas imágenes cuyo valor SSIM fuera menor a 0.75.

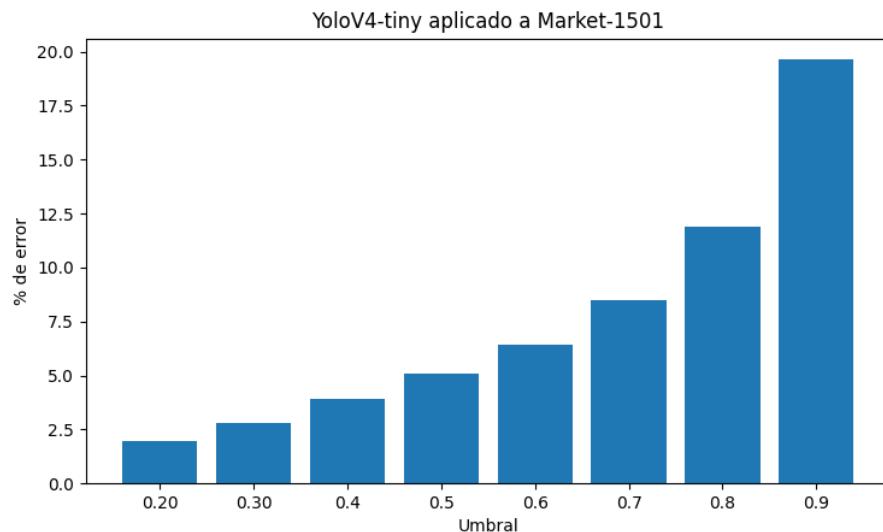


Figura 5.3: Aplicación de YoloV4 tiny sobre las imágenes de la base de datos Market-1501 utilizando diferentes número de umbrales. El porcentaje de error son las imágenes que no ha clasificado como peatón.

Método	Market-1501 FID	Referencia
Real	7.22	Hao Chen <i>et al.</i> (42)
IS-GAN	281.63	Hao Chen <i>et al.</i> (42)
FD-GAN	257.00	Saleh Hussin <i>et al.</i> (48)
PG-GAN	151.16	Zhedong Zheng <i>et al.</i> (7)
DCGAN	136.26	Saleh Hussin <i>et al.</i> (48)
LSGAN	136.26	Zhedong Zheng <i>et al.</i> (7)
PN-GAN	54.23	Zhedong Zheng <i>et al.</i> (7)
GCL	53.07	Hao Chen <i>et al.</i> (42)
DG-Net	18.24	Hao Chen <i>et al.</i> (42)
DG-GAN	18.24	Saleh Hussin <i>et al.</i> (48)
StyleGAN3	9.29	

Tabla 5.4: Tabla comparativa de diferentes redes generativas adversarias. Utilizando la métrica FID (60) y entrenando todos los modelos con la base de datos Market-1501 (1). La primera fila es el valor obtenido al aplicar la métrica a las imágenes reales de la base de datos

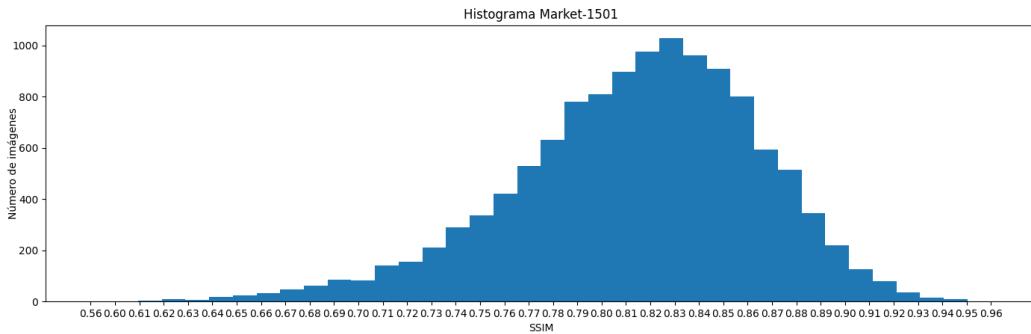


Figura 5.4: Histograma de la métrica SSIM (9) sobre la base de datos Market-1501. Número de imágenes que han obtenido el mismo valor SSIM. Se seleccionó una imagen por persona y se comparó con el resto de imágenes de esa misma persona. Cómo se puede observar la mayoría de las imágenes rondan el umbral 0.75 en adelante.



Figura 5.5: A) Imágenes reales de la base de datos Market-1501. B) Imágenes artificiales.

a) Generación de personas artificiales

Durante la experimentación se generaron de maneras totalmente aleatoria las imágenes de 401 personas artificiales y mediante la modificación de sus vectores latentes se generaron 51 imágenes por persona en diferentes posturas, haciendo un total de 20,451 imágenes (ver Fig. 5.6).

A las imágenes generadas se aplicó el filtro Yolo V4 para la detección de peatones eliminando 3,419 imágenes que representan un 16.7 % del total, en la Fig. 5.7 se muestran diferentes ejemplos de imágenes filtradas.

Después se aplicó el filtro SSIM y se descartaron 386, un 2.3 % de las imágenes. En la Fig. 5.8 se pueden ver algunos ejemplos de imágenes descartadas con este método.

Una vez aplicados los filtros se descartaron un total de 3,815 imágenes (ver Tabla 5.5).



Figura 5.6: Semilla - es la imagen generada de manera aleatoria a la cual se le van a modificar sus vectores latentes para modificar sus características medias. Generadas - son las imágenes que se han generado al modificar los vectores latentes de la imagen semilla.

Método	Imgs. descartadas	%
Yolov4-tiny pedestrian detection (59)	3.419	16.7
SSIM (9)	396	2.3
TOTAL	3.815	18.6

Tabla 5.5: Número de imágenes descartadas durante la aplicación de diferentes filtros.



Figura 5.7: Ejemplo de algunas imágenes descartadas utilizando el modelo Yolo V4 tiny para detectar peatones.



Figura 5.8: Ejemplo de algunas imágenes descartadas utilizando la métrica SSIM (9). Partiendo como base de una de las imágenes de una persona (imagen original), se compara con el resto de imágenes generadas de esa misma persona.

b) Generación de imágenes de personas reales

En el contexto de la generación de imágenes de personas reales, se utiliza un encoder que ha sido reentrenado con el modelo StyleGAN3 preentrenado con las base de datos Market-1501. Este encoder es el mismo utilizado en el artículo presentado por Yuval Alaluf *et al.* (57), el cual ha demostrado ser altamente efectivo en la generación de imágenes con calidad visual muy similar a las de caras de personas reales. El objetivo de utilizar este encoder es poder codificar imágenes de personas reales y encontrar sus vectores latentes correspondientes dentro del generador de StyleGAN3, lo que permitirá generar variaciones de la imagen original a través de manipulaciones en los vectores latentes.

Para la generación de imágenes de una persona real se necesitó entrenar el codificador durante 240,000 épocas (ver Tabla 5.6).

Modelo	Img. entrenamiento	Img. validación	Entrenamiento	Épocas	Hardware
stylegan3-editing	39466	732	3d 03h 16m	240,000	Titan RTX

Tabla 5.6: Datos técnicos del entrenamiento del modelo para la generación de imágenes artificiales.

Para evaluar el rendimiento del modelo durante el entrenamiento se utilizaron tres funciones de pérdida distintas, que permiten medir diferentes aspectos de la calidad de las imágenes generadas. La primera de ellas es la Perceptual Similarity Metric (61) (LPIPS), que mide la similitud perceptual entre dos imágenes. La segunda es la L2 (62), que mide la diferencia euclíadiana entre dos imágenes. Por último, se utilizó la función de pérdida Momentum Contrast (63) (MOCO), que tiene en cuenta la correlación entre las características de diferentes imágenes.

En la Tabla 5.7 se puede observar cómo fluctúan las tres funciones de pérdida en las diferentes épocas del entrenamiento. Es importante tener en cuenta que el rendimiento del modelo puede variar en función de la complejidad de los datos de entrada, y que la arquitectura de StyleGAN3 está diseñada originalmente para trabajar con conjuntos de datos más simples, como caras. Es por eso que el rendimiento del modelo puede verse afectado cuando se utilizan imágenes más complejas, como el cuerpo completo de una persona en diferentes posturas.

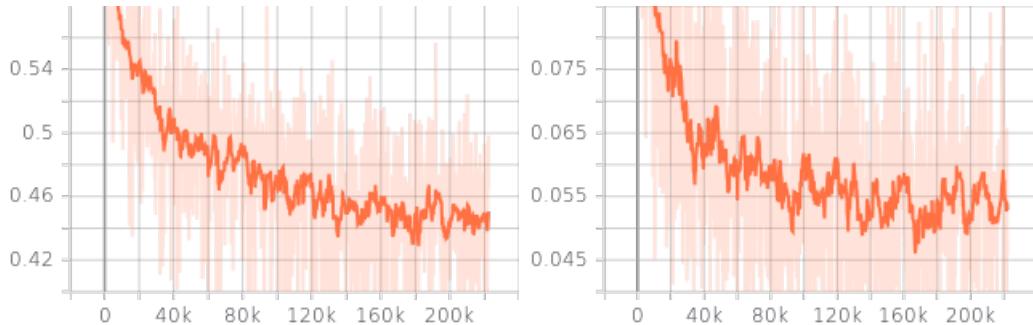


Figura 5.9: LPIPS

Figura 5.10: L2

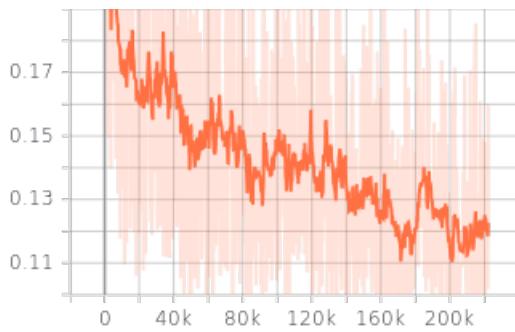


Figura 5.11: MOCO

Tabla 5.7: Evolución de las funciones de pérdida durante el entrenamiento hasta la época 220,000.

Seguidamente se observa como en la Tabla 5.8 el aprendizaje es más suave y se puede observar como la métrica LPIPS es la que va mejorando de manera más estable.

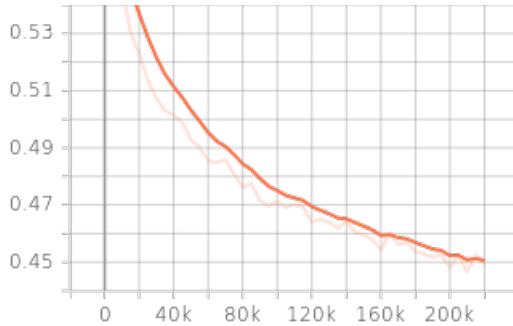


Figura 5.12: LPIPS

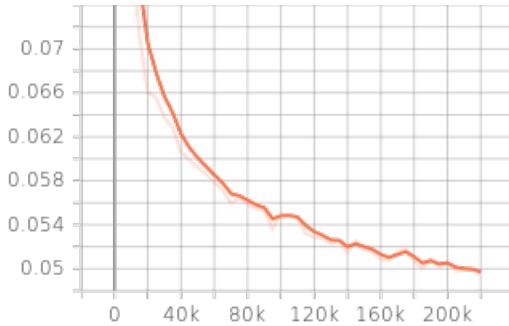


Figura 5.13: L2

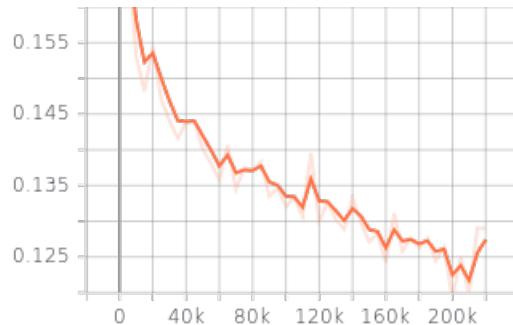


Figura 5.14: MOCO

Tabla 5.8: Conjunto de funciones de pérdida durante la validación en las distintas épocas, hasta la época 220,000.

Utilizando el modelo generado en la época 220,000 se obtuvieron los vectores latentes que representan las imágenes de entrada. Como se puede observar en la Fig. 5.16, las imágenes obtenidas se asemejan a las originales, aunque no alcanzan la calidad de las imágenes generadas en el punto anterior.

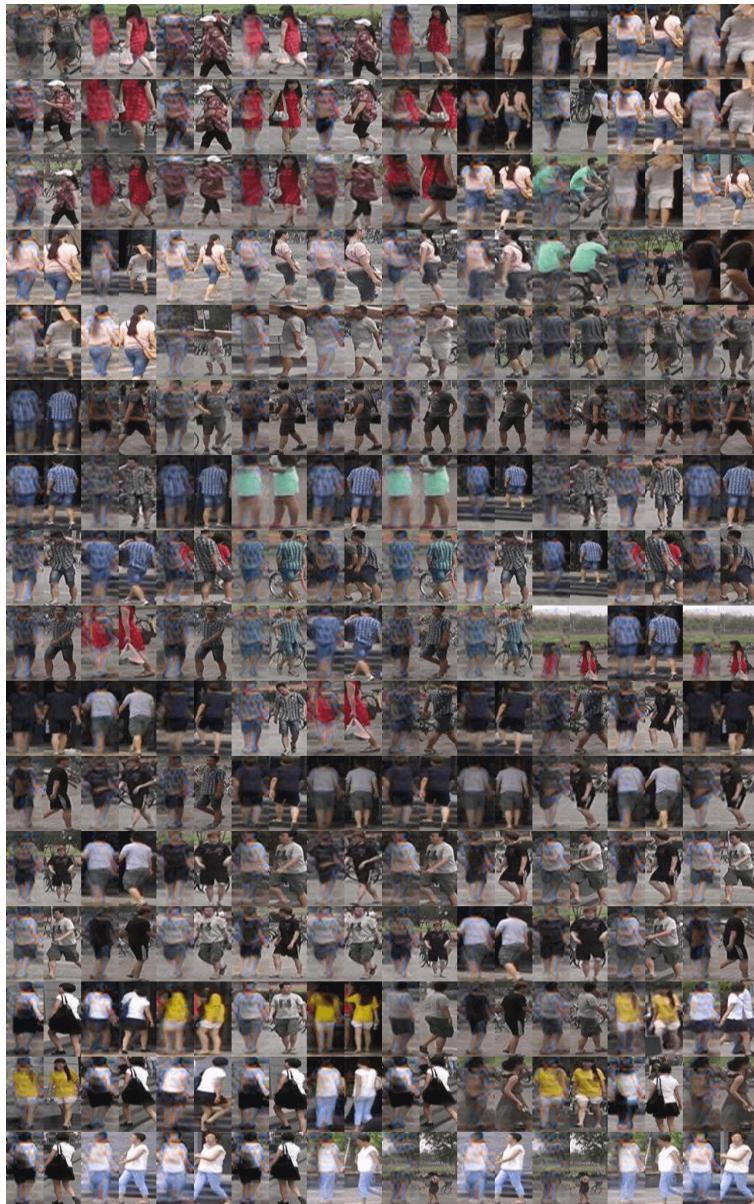


Figura 5.15: Pares de imágenes. Derecha se encuentra la imagen real, y a la izquierda su homólogo obtenido a través del codificador dentro del espacio latente de StyleGAN3. Se puede observar que el modelo funciona mejor cuando se muestra el cuerpo completo, aunque no alcanza la calidad de las imágenes generadas de manera aleatoria.

Método	Img. descartadas	%
Yolov4-tiny pedestrian detection (59)	12,129	16.0
SSIM (9)	1,108	1.46
TOTAL	13,237	17.46

Tabla 5.9: Número de imágenes descartadas durante la aplicación de diferentes filtros.

Una vez que se ha generado el vector latente se procede a generar las variantes de esa persona en diferentes posturas modificando los vectores latentes. Se generaron un total de 75,788 imágenes de 751 personas diferentes. Despues de aplicar los diferentes filtros quedaron 63,659 imágenes como se muestra en la Tabla 5.9



Figura 5.16: Semilla - imagen real. Generadas - son las imágenes que se han generado al modificar los vectores latentes de la imagen semilla.

5.2. Re-identificación

Se hicieron dos tipos de pruebas durante la experimentación en función del tipo de imágenes artificiales generadas. En primer lugar se analiza el rendimiento del modelo cuando se añaden imágenes de personas artificiales y sus variantes en diferentes posturas y posteriormente con imágenes artificiales generadas a partir de imágenes de personas reales. Para el entrenamiento como hiperparámetros se utilizó un tamaño de lote, `-batchsize` de 16 imágenes, se utilizó sólo una GPU y en todas las diferentes pruebas se entrenó el modelo durante 60 épocas.

a) Utilizando 320 personas generadas de manera artificial.

Durante la experimentación se probó el modelo con un número diferentes de personas añadidas, se fueron añadiendo de diez en diez hasta llegar a trescientas veinte. En la Fig. 5.21 se puede ver como el rendimiento del modelo de re-identificación base se queda estable o hasta baja un poco y luego empieza a subir, esto puede ser debido a que al aumentar el número de personas también se aumenta el número de clases y debido a que tienen diferente número de imágenes puede ser que alguna clases no sean tan relevantes como otras y el añadir más sólo está generando un pequeño ruido. Llegando a mejorar un 1 % al añadir 280 personas durante el entrenamiento.

Durante el entrenamiento se puede observar como a partir de la época 40 se estabiliza el modelo de igual manera en todos los experimentos, no habiendo un cambio significativo en función del número de personas añadidas, (ver Figs. 5.23, 5.24, 5.25, 5.20)

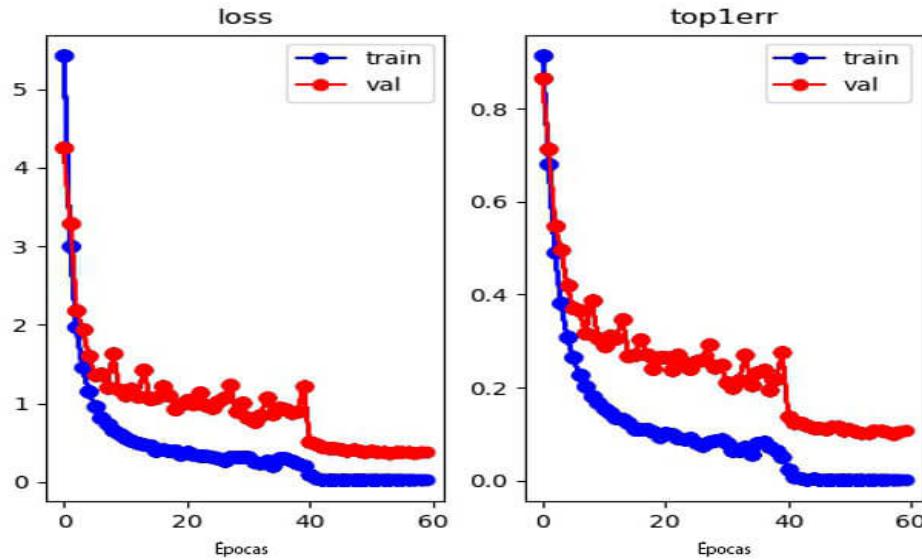


Figura 5.17: Base, sin personas añadidas. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

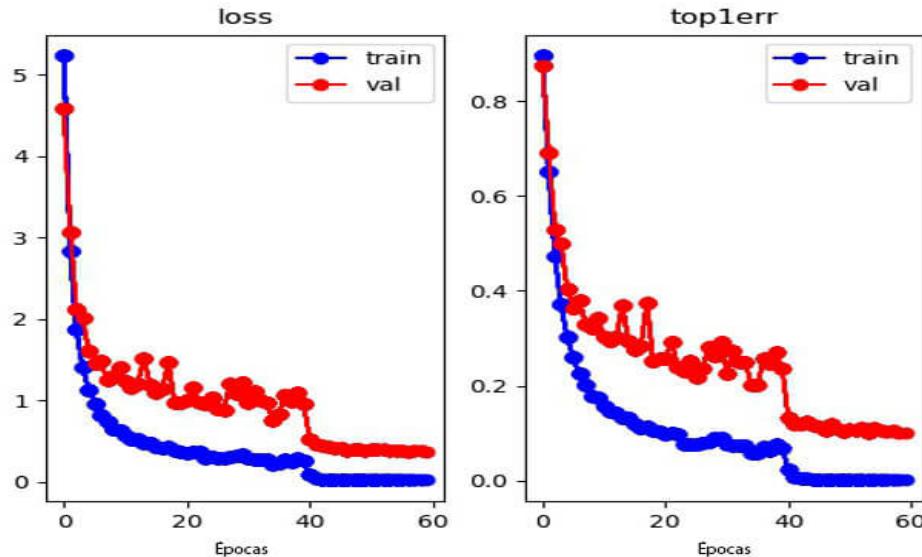


Figura 5.18: Añadiendo 100 personas. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

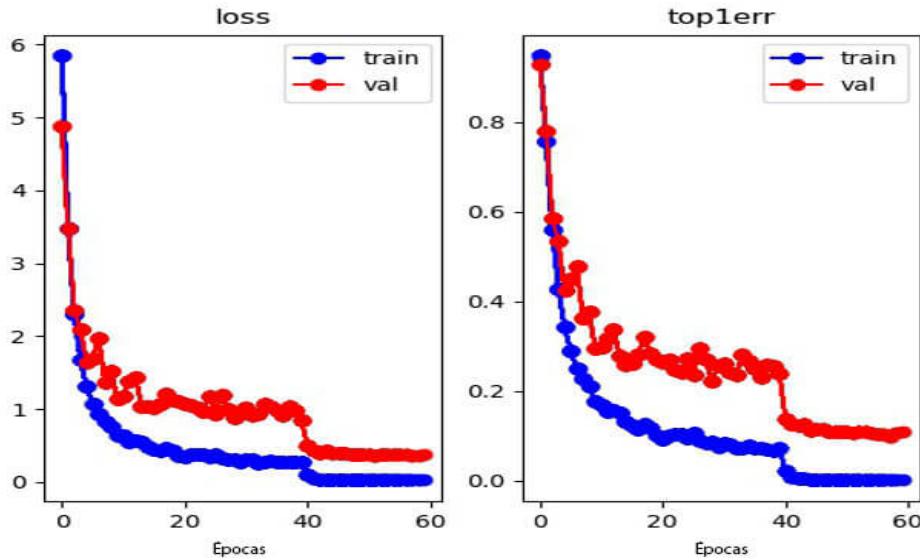


Figura 5.19: Añadiendo 200 personas. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

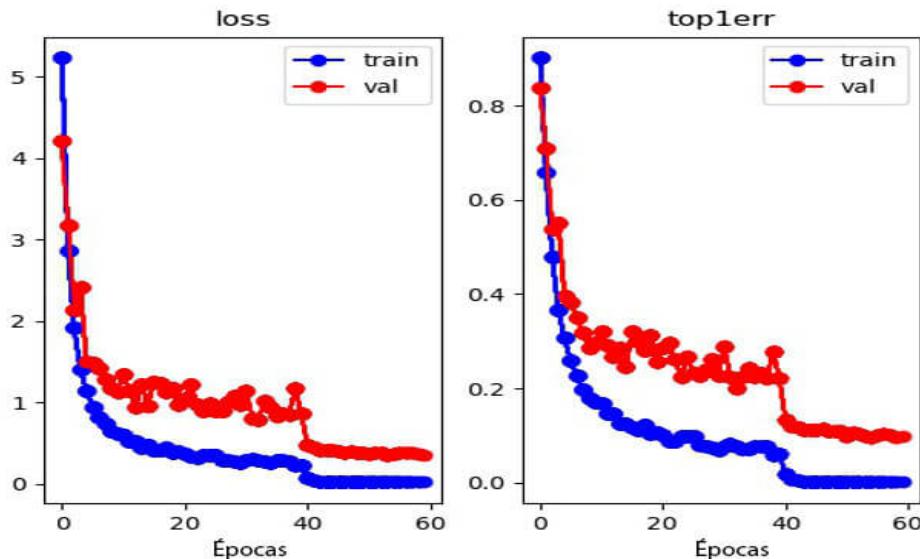


Figura 5.20: Añadiendo 320 personas. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

Pers.	Img.	Rank
0	0	Rank@1:0.893112 Rank@5:0.964964 Rank@10:0.977732 mAP:0.742632
10	473	Rank@1:0.888955 Rank@5:0.964074 Rank@10:0.980404 mAP:0.743795
20	874	Rank@1:0.892221 Rank@5:0.961105 Rank@10:0.974169 mAP:0.745116
30	1252	Rank@1:0.891627 Rank@5:0.965558 Rank@10:0.979513 mAP:0.741414
40	1.682	Rank@1:0.890143 Rank@5:0.964371 Rank@10:0.979513 mAP:0.748799
50	2.046	Rank@1:0.894299 Rank@5:0.962589 Rank@10:0.978028 mAP:0.746853
60	2.427	Rank@1:0.901128 Rank@5:0.967637 Rank@10:0.980404 mAP:0.751229
70	2.859	Rank@1:0.892815 Rank@5:0.965261 Rank@10:0.978325 mAP:0.748843
80	3.214	Rank@1:0.892518 Rank@5:0.965261 Rank@10:0.977732 mAP:0.748257
90	3647	Rank@1:0.899347 Rank@5:0.964964 Rank@10:0.979216 mAP:0.758927
100	4058	Rank@1:0.898159 Rank@5:0.964667 Rank@10:0.980998 mAP:0.755366
150	6.167	Rank@1:0.898753 Rank@5:0.965261 Rank@10:0.979513 mAP:0.761433
200	8.223	Rank@1:0.896378 Rank@5:0.967340 Rank@10:0.980701 mAP:0.763768
250	10.307	Rank@1:0.893705 Rank@5:0.964964 Rank@10:0.980107 mAP:0.762484
280	11.244	Rank@1:0.903504 Rank@5:0.966746 Rank@10:0.982185 mAP:0.767955
300	12.320	Rank@1:0.896081 Rank@5:0.963777 Rank@10:0.978919 mAP:0.768727
320	14.371	Rank@1:0.896675 Rank@5:0.965855 Rank@10:0.978622 mAP:0.769509

Tabla 5.10: Resultados entrenamiento añadiendo un número diferente de personas. La primera fila es la base, sin añadir ninguna imagen.

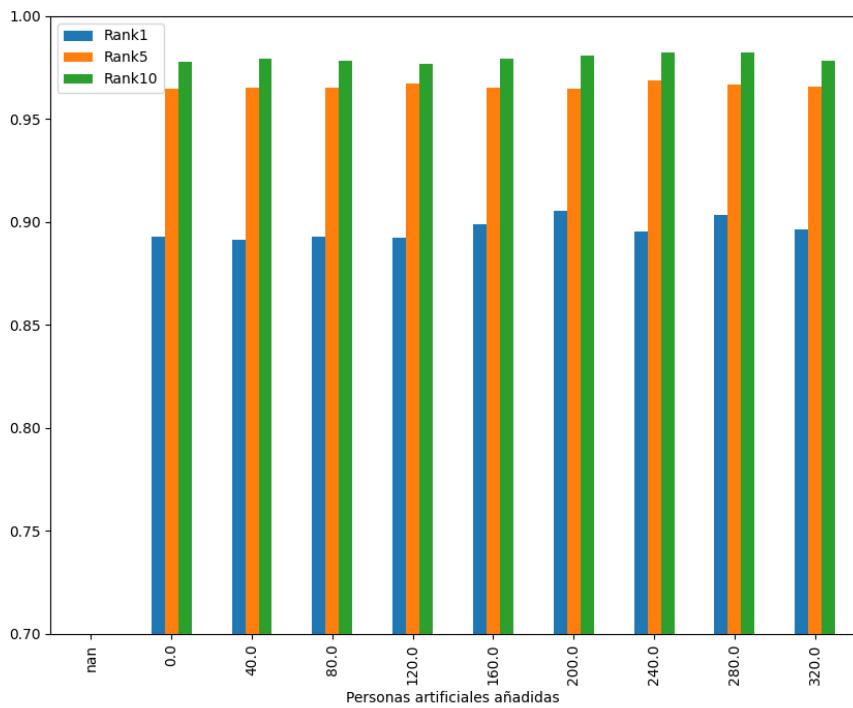


Figura 5.21: Rendimiento de algunos modelos entrenados con diferente número de personas generadas de manera artificial (ver Tabla 5.10). Añadiendo 0, 40, 80, 120, 160, 200, 240, 280 y 320 personas

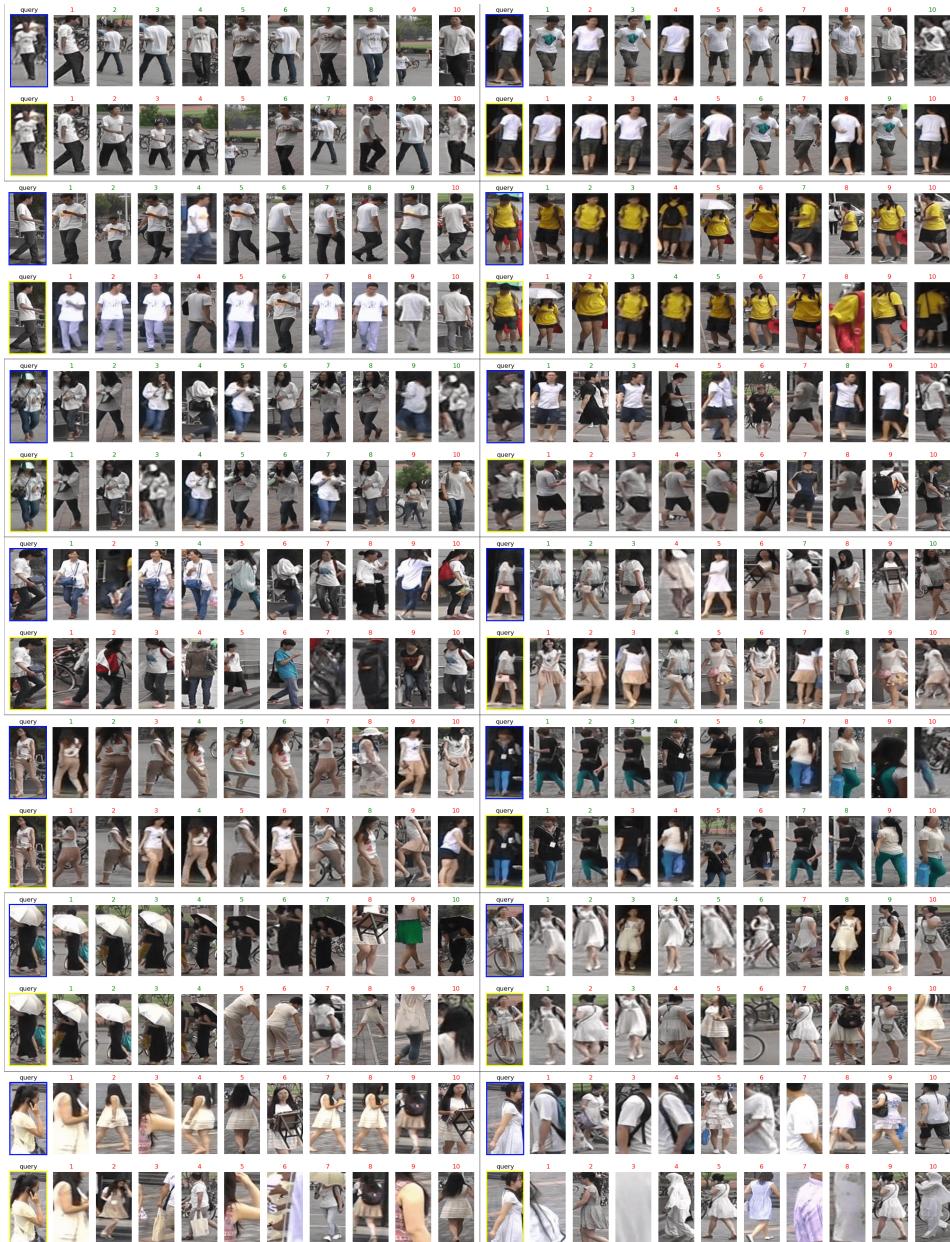


Figura 5.22: Resultados modelo de re-identificación. Comparación de resultados con el modelo base (■) y modelo después de añadir 280 personas (■). Query es la imagen de la persona a buscar, y las siguientes imágenes representan la salida del modelo, el color verde es un acierto y rojo un error.

b) Agregando imágenes artificiales en diferentes posturas a cada persona real.

Durante la experimentación se probó añadiendo más imágenes a cada una de las personas del lote de entrenamiento, en cada prueba se fueron añadiendo de cinco en cinco. En la Fig. 5.21 se puede ver como el rendimiento del modelo de re-identificación base va empeorando a medida que le vamos añadiendo imágenes, llegando a bajar el rendimiento un 8% al añadir 100 imágenes por persona, esto es debido a que la calidad de las imágenes generadas no ha sido buena y se muestran muy difusas con respecto a las imágenes reales de entrenamiento.

De igual manera que en el apartado anterior durante el entrenamiento se puede observar como a partir de la época 40 se estabiliza el modelo de igual manera en todos los experimentos, no habiendo un cambio significativo en función del número de imágenes por persona añadidas, (ver Fig. 5.23, 5.24, 5.25, 5.20).

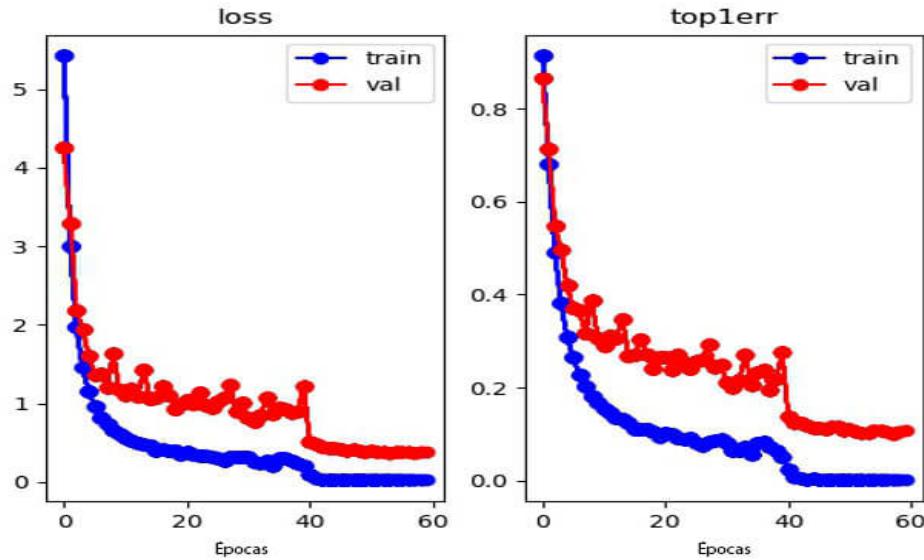


Figura 5.23: Base, sin personas añadidas. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

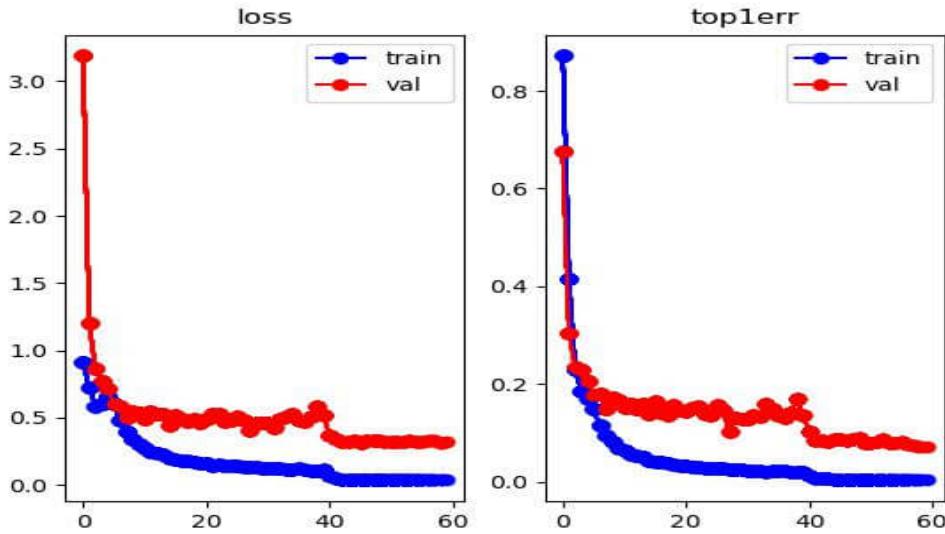


Figura 5.24: Añadiendo 35 imágenes a cada persona. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

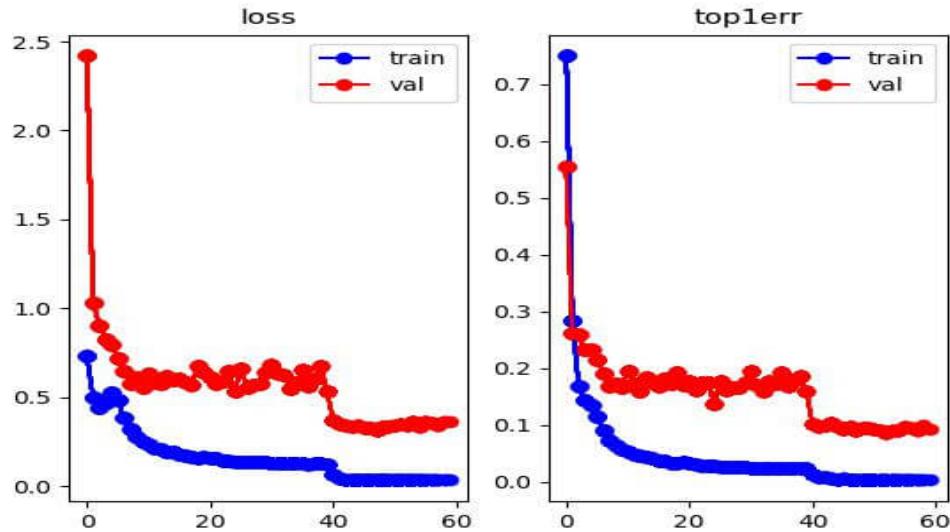


Figura 5.25: Añadiendo 100 imágenes a cada persona. **Izquierda**, función de pérdida durante el entrenamiento y validación. **Derecha**, porcentaje de error en Rank1 durante el entrenamiento y validación.

Pers.	Img.	Rank
0	0	Rank@1:0.893112 Rank@5:0.964964 Rank@10:0.977732 mAP:0.742632
751	5	Rank@1:0.886876 Rank@5:0.959620 Rank@10:0.975950 mAP:0.719692
751	10	Rank@1:0.878860 Rank@5:0.958432 Rank@10:0.977138 mAP:0.704901
751	15	Rank@1:0.870546 Rank@5:0.958729 Rank@10:0.969715 mAP:0.686172
751	20	Rank@1:0.865796 Rank@5:0.950119 Rank@10:0.972981 mAP:0.674047
751	25	Rank@1:0.861342 Rank@5:0.955166 Rank@10:0.971793 mAP:0.663773
751	30	Rank@1:0.850950 Rank@5:0.951010 Rank@10:0.969121 mAP:0.654492
751	35	Rank@1:0.845606 Rank@5:0.940915 Rank@10:0.964074 mAP:0.635790
751	40	Rank@1:0.841449 Rank@5:0.944477 Rank@10:0.966746 mAP:0.640035
751	45	Rank@1:0.826306 Rank@5:0.935273 Rank@10:0.961105 mAP:0.630469
751	60	Rank@1:0.829869 Rank@5:0.938836 Rank@10:0.960214 mAP:0.624382
751	70	Rank@1:0.832245 Rank@5:0.942102 Rank@10:0.965855 mAP:0.623470
751	85	Rank@1:0.825713 Rank@5:0.935570 Rank@10:0.959620 mAP:0.614014
751	105	Rank@1:0.817399 Rank@5:0.932304 Rank@10:0.958729 mAP:0.605653
751	135	Rank@1:0.813539 Rank@5:0.931710 Rank@10:0.958135 mAP:0.601084

Tabla 5.11: Resultados entrenamiento añadiendo un número diferente de personas. La primera fila es la base, sin añadir ninguna imagen.

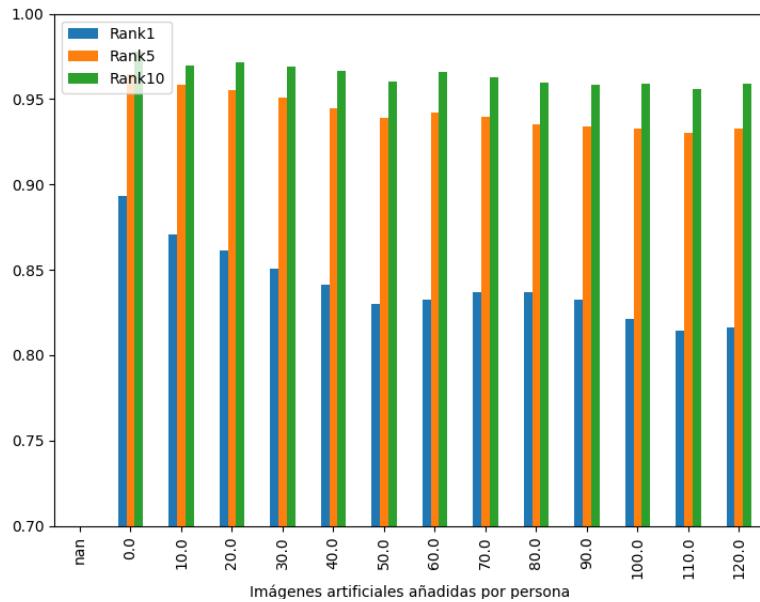


Figura 5.26: Rendimiento de algunos modelos entrenados con diferente número de imágenes artificiales por persona (ver Tabla 5.11). Añadiendo 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 y 120 imágenes por persona.

Capítulo 6

Conclusiones y trabajo a futuro

El uso de redes generativas adversarias para aumentar datos es una técnica prometedora para mejorar el rendimiento en modelos de re-identificación. Los resultados obtenidos en nuestro experimento demuestran que esta técnica es efectiva en la generación de datos de alta calidad y la versatilidad de generar modificaciones de las mismas.

Cómo resultado de la experimentación se pudo observar que añadiendo personas totalmente artificiales el modelo de re-identificación pudo mejorar su rendimiento un 1 %. No se pudo lograr lo mismo con el aumento de las imágenes en personas existentes debido a que el codificador no logró obtener los vectores latentes semejantes a las imágenes reales, esto puede ser debido a que estas herramientas están preparadas para funcionar con caras no con imágenes tan complejas como un cuerpo entero.

El uso de estas redes generativas adversarias permite la utilización de menos datos originales en los entrenamientos, lo que puede reducir los requerimientos de recursos y tiempo en el proceso de entrenamiento de los modelos. En resumen, el uso de redes generativas adversarias en el ámbito de la re-identificación es una técnica valiosa que puede aportar una mejora significativa en el rendimiento de los modelos de re-identificación. Permiten adaptarse a diferentes conjuntos de datos y situaciones. Esto hace que sea una herramienta valiosa no solo para el campo de la re-identificación, sino también para otros campos donde se requiera la generación y mejora de datos.

Como trabajo a futuro se podría hacer *fine-tuning* al codificador para que mejore su rendimiento generando los vectores latentes. También se podría mejorar el rendimiento aplicando otros filtros que que eliminen las imágenes de peor calidad. Debido a la gran cantidad de las imágenes, este proceso debe ser automático. Otra opción sería generar los datos mediante plantillas de posturas, es decir, entrenando Stylegan3 de manera supervisada con imágenes de personas en diferentes posturas. Se podrían utilizar otras redes generativas adversarias, pero para este caso, considero que Stylegan ha generado imágenes artificiales de una gran calidad y diversidad. Otra posibilidad sería cambiar el paradigma de los modelos de re-identificación, actualmente está basada en la red neuronal Resnet50 con leves modificaciones, se utiliza como extractor de características. Una propuesta interesante sería la utilización de los vectores latentes en Stylegan3. Es decir medir la distancia coseno que tienen las imágenes, sus vectores latentes, dentro de Stylegan3. Pero para ello se necesita que el codificador esté funcionando de manera correcta.

Actualmente nos encontramos con un auge de contenido generado de manera artificial, cómo son los modelos de difusión o en inglés ("*diffusion model*"). Un modelo de difusión es un tipo de modelo matemático que se puede utilizar para generar datos artificiales. Este modelo se basa en la ecuación de difusión, que es una ecuación diferencial que describe cómo una cantidad se dispersa en un medio continuo. La idea es que si se puede modelar la difusión de una cantidad en un medio continuo, entonces se pueden generar datos artificiales que sean similares a los datos reales.

En el contexto de la re-identificación, se podría utilizar un modelo de difusión para generar datos artificiales de personas en diferentes escenas. Estos datos artificiales se podrían utilizar para entrenar un modelo de re-identificación, lo que podría aumentar su rendimiento. Por ejemplo el modelo de código abierto Stable Diffusion ha sido un parte aguas en la generación de imágenes artificiales y podría ser muy interesante ver como se puede comportar para la generación de imágenes de personas.

Como se puede observar, existen diversas formas de abordar y solucionar el problema, y gracias a los avances en las arquitecturas de los modelos generativos, es posible plantear una amplia variedad de soluciones para mejorar el rendimiento de cualquier modelo que requiera aumentar sus datos de entrenamiento de forma artificial.

Bibliografía

- [1] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable Person Re-identification : A Benchmark University of Texas at San Antonio,” *Iccv*, pp. 1116–1124, 2015. [Online]. Available: <http://www.liangzheng.com.cn>.
- [2] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, “Alias-Free Generative Adversarial Networks,” no. NeurIPS, 2021. [Online]. Available: <http://arxiv.org/abs/2106.12423>
- [3] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, “CamStyle: A Novel Data Augmentation Method for Person Re-Identification,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1176–1190, 2019.
- [4] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9914 LNCS, no. c, pp. 17–35, 2016.
- [5] W. Li, R. Zhao, T. Xiao, and X. Wang, “DeepReID: Deep filter pairing neural network for person re-identification,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 152–159, 2014.
- [6] D. Gray, S. Brennan, and H. Tao, “Evaluating appearance models for recognition, reacquisition, and tracking,” *10th International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*,, vol. 3, no. March, pp. 41–47, 2007.
- [7] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, “Joint discriminative and generative learning for person re-identification,” *Pro-*

- ceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 2133–2142, 2019.
- [8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8107–8116, 2020.
 - [9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
 - [10] C. Li, K. Xu, J. Zhu, and B. Zhang, “Triple generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 2014-Decem, pp. 4089–4099, 2014.
 - [11] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2242–2251, 2017.
 - [12] P. Dai, R. Ji, H. Wang, Q. Wu, and Y. Huang, “Cross-modality person re-identification with generative adversarial training,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, pp. 677–683, 2018.
 - [13] A. Wu, W.-s. Zheng, H.-x. Yu, S. Gong, and J. Lai, “RGB-Infrared Cross-Modality Person Re-Identification,” pp. 5380–5389.
 - [14] W. Liang, G. Wang, J. Lai, and J. Zhu, “M2M-GAN: Many-to-Many Generative Adversarial Transfer Learning for Person Re-Identification,” 2018. [Online]. Available: <http://arxiv.org/abs/1811.03768>
 - [15] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, “Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-identification,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 994–1003, 2018.
 - [16] A. H. Re-identification, “arXiv : 1607 . 08378v2 [cs . CV] 26 Sep 2016,” pp. 1–18.

- [17] S. Zhou, M. Ke, and P. Luo, “Multi-camera transfer GAN for person re-identification,” *Journal of Visual Communication and Image Representation*, vol. 59, pp. 393–400, 2019. [Online]. Available: <https://doi.org/10.1016/j.jvcir.2019.01.029>
- [18] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.
- [19] Y. Tang, Y. Xi, N. Wang, B. Song, and X. Gao, “CGAN-TM: A Novel Domain-to-Domain Transferring Method for Person Re-Identification,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5641–5651, 2020.
- [20] Y. Khraimeche, G.-A. Bilodeau, D. Steele, and H. Mahadik, “Unsupervised Disentanglement GAN for Domain Adaptive Person Re-Identification,” 2020. [Online]. Available: <http://arxiv.org/abs/2007.15560>
- [21] R. Sun, W. Lu, Y. Zhao, J. Zhang, and C. Kai, “A Novel Method for Person Re-Identification: Conditional Translated Network Based on GANs,” *IEEE Access*, vol. 8, pp. 3677–3686, 2020.
- [22] G. Wang, Y. Y. Yang, J. Cheng, J. Wang, and Z. Hou, “Color-sensitive person re-identification,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, no. May 2020, pp. 933–939, 2019.
- [23] C. Liu, X. Chang, and Y. D. Shen, “Unity style transfer for person re-identification,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6886–6895, 2020.
- [24] T. Kim, M. Cha, H. Kim, J. Kwon, and L. Jiwon, “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks.”
- [25] X. Zhang, X. Y. Jing, X. Zhu, and F. Ma, “Semi-supervised person re-identification by similarity-embedded cycle GANs,” *Neural Computing and Applications*, vol. 32, no. 17, pp. 14 143–14 152, 2020. [Online]. Available: <https://doi.org/10.1007/s00521-020-04809-7>

- [26] Z. Pang, J. Guo, W. Sun, Y. Xiao, and M. Yu, “Cross-domain person re-identification by hybrid supervised and unsupervised learning,” *Applied Intelligence*, 2021.
- [27] K. He, “Deep Residual Learning for Image Recognition.”
- [28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, C. V. Jan, J. Krause, and S. Ma, “ImageNet Large Scale Visual Recognition Challenge.”
- [29] Y. Li, S. Chen, G. Qi, Z. Zhu, M. Haner, and R. Cai, “Imaging A GAN-Based Self-Training Framework for Unsupervised Domain Adaptive Person Re-Identification,” pp. 1–16, 2021. [Online]. Available: <https://www.mdpi.com/2313-433X/7/4/62>
- [30] X. Luo, Z. Ouyang, N. Du, J. Song, and Q. Wei, “Cross-Domain Person Re-Identification Based on Feature Fusion,” *IEEE Access*, vol. 9, pp. 98 327–98 336, 2021.
- [31] P. Re-identification, “Pose-Normalized Image Generation for,” *The European Conference on Computer Vision (ECCV)*, pp. 650–667, 2018. [Online]. Available: http://openaccess.thecvf.com/content_ECCV_2018/html/Xuelin_Qian_Pose-Normalized_Image_Generation_ECCV_2018_paper.html
- [32] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1302–1310, 2017.
- [33] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, “Deformable GANs for Pose-Based Human Image Generation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3408–3416, 2018.
- [34] Y. Li, T. Zhang, L. Duan, and C. Xu, “A unified generative adversarial framework for image generation and person re-identification,” *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pp. 163–172, 2018.

- [35] Y. Ge, Z. Li, H. Zhao, G. Yin, S. Yi, X. Wang, and H. Li, “FD-GAN: Pose-guided Feature Distilling GAN for Robust Person Re-identification,” *Advances in Neural Information Processing Systems*, vol. 2018-Decem, no. NeurIPS, pp. 1222–1233, 2018.
- [36] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.
- [37] N. M. G.-b. P.-a. R. Video-based, A. Borgia, Y. Hua, E. Kodirov, and N. M. Robertson, “GAN-based Pose-aware Regulation for Video-based Person Re- identification GAN-based Pose-aware Regulation for Video-based Person Re-identification,” 2019.
- [38] C. Zhang, L. Zhu, S. C. Zhang, and W. Yu, “PAC-GAN: An effective pose augmentation scheme for unsupervised cross-view person re-identification,” *Neurocomputing*, vol. 387, pp. 22–39, 2020. [Online]. Available: <https://doi.org/10.1016/j.neucom.2019.12.094>
- [39] C. Zhang, L. Wu, and Y. Wang, “Crossing generative adversarial networks for cross-view person re-identification,” *Neurocomputing*, vol. 340, pp. 259–269, 2019.
- [40] Y. Zhang, Y. Jin, J. Chen, S. Kan, Y. Cen, and Q. Cao, “PGAN: Part-based nondirect coupling embedded gan for person reidentification,” *IEEE Multimedia*, vol. 27, no. 3, pp. 23–33, 2020.
- [41] Z. Ni, J. Pei, and Y. Zhao, “Affine transform for skew correction based on generative adversarial network method for multi-camera person re-identification,” *ACM International Conference Proceeding Series*, vol. PartF16898, pp. 89–95, 2021.
- [42] H. Chen, Y. Wang, B. Lagadec, A. Dantcheva, and F. Bremond, “Joint Generative and Contrastive Learning for Unsupervised Person Re-identification,” pp. 2004–2013, 2020. [Online]. Available: <http://arxiv.org/abs/2012.09071>
- [43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proceedings of the*

- IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016.
- [44] Z. Zheng, L. Zheng, and Y. Yang, “Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 3774–3782, 2017.
 - [45] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.
 - [46] Y. Huang, J. Xu, Q. Wu, Z. Zheng, Z. Zhang, and J. Zhang, “Multi-pseudo regularized label for generated data in person re-identification,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1391–1403, 2019.
 - [47] J. P. Ainam, K. Qin, G. Liu, and G. Luo, “Sparse Label Smoothing Regularization for Person Re-Identification,” *IEEE Access*, vol. 7, pp. 27 899–27 910, 2019.
 - [48] S. H. S. Hussin and R. Yildirim, “StyleGAN-LSRO Method for Person Re-identification,” *IEEE Access*, pp. 13 857–13 869, 2021.
 - [49] C. Eom and B. Ham, “Learning disentangled representation for robust person re-identification,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
 - [50] L. Xia, J. Zhu, and Z. Yu, “Real-World Person Re-Identification via Super-Resolution and Semi-Supervised Methods,” *IEEE Access*, vol. 9, pp. 35 834–35 845, 2021.
 - [51] H. Alqahtani, M. Kavakli-Thorne, and C. Z. Liu, “An introduction to person re-identification with generative adversarial networks,” *arXiv*, pp. 1–15, 2019.
 - [52] Z. Luo, “Review of GAN-Based Person Re-Identification,” 2021.
 - [53] Y. Jiang, W. Chen, X. Sun, X. Shi, F. Wang, and H. Li, *Exploring the Quality of GAN Generated Images for Person Re-Identification*. Association for Computing Machinery, 2021, vol. 1, no. 1.

- [54] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 4396–4405, 2019.
- [55] Q. Cai and J. K. Aggarwal, “Tracking Human Motion Using Multiple Cameras,” 1996.
- [56] P. DImitrakopoulos, G. Sfikas, and C. Nikou, “Wind: Wasserstein Inception Distance for Evaluating Generative Adversarial Network Performance,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, pp. 3182–3186, 2020.
- [57] Y. Alaluf, O. Patashnik, Z. Wu, A. Zamir, E. Shechtman, D. Lischinski, and D. Cohen-Or, “Third Time’s the Charm? Image and Video Editing with StyleGAN3,” 2022. [Online]. Available: <http://arxiv.org/abs/2201.13433>
- [58] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, “Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2287–2296, 2021.
- [59] Z. Jiang, L. Zhao, L. I. Shuaiyang, and J. I. Yanfei, “Real-time object detection method for embedded devices,” *arXiv*, vol. 3, no. October, pp. 1–11, 2020.
- [60] Y. Yu, D. Zhang Weibin, and Yun, “Frechet Inception Distance (FID) for Evaluating GANs,” no. September, pp. 0–7, 2021.
- [61] M. Kettunen, E. Häkkinen, and J. Lehtinen, “E-LPIPS: Robust Perceptual Image Similarity via Random Transformation Ensembles,” 2019. [Online]. Available: <http://arxiv.org/abs/1906.03973>
- [62] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss Functions for Neural Networks for Image Processing,” no. November, 2015. [Online]. Available: <http://arxiv.org/abs/1511.08861>
- [63] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” *Proceedings of the*

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 9726–9735, 2020.