

**MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS**



VNIVERSITAT  
E VALÈNCIA

**TRABAJO DE FIN DE MÁSTER**

**GENERACIÓN DE IMÁGENES SINTÉTICAS DE  
RESONANCIA MAGNÉTICA CARDÍACA MEDIANTE  
TÉCNICAS DE APRENDIZAJE PROFUNDO**

**AUTOR:**

**JOSE MANUEL JAÉN LORITES**

**TUTORES:**

**VALERO LAPARRA PÉREZ-MUELAS**

**MANUEL PÉREZ PELEGRÍ**

**DAVID MORATAL PÉREZ**

**ENERO, 2022**





VNIVERSITAT  
E VALÈNCIA



Escola Tècnica Superior  
d'Enginyeria **ETSE-UV**

## MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS

### TRABAJO DE FIN DE MÁSTER

# GENERACIÓN DE IMÁGENES SINTÉTICAS DE RESONANCIA MAGNÉTICA CARDÍACA MEDIANTE TÉCNICAS DE APRENDIZAJE PROFUNDO

#### AUTOR:

**JOSE MANUEL JAÉN LORITES**

#### TUTORES:

**VALERO LAPARRA PÉREZ-MUELAS**

**MANUEL PÉREZ PELEGRÍ**

**DAVID MORATAL PÉREZ**

---

#### TRIBUNAL:

PRESIDENTE/A:

VOCAL 1:

VOCAL 2:

**FECHA DE DEFENSA:**

**CALIFICACIÓN:**



## Resumen

Las técnicas de inteligencia artificial han demostrado tener un gran potencial en el área de la cardiología. Debido a las dificultades que supone el trabajo con imágenes de pacientes reales, cada vez más se están utilizando técnicas para generar imágenes sintéticas con el mayor grado de similitud a las reales, a partir de las cuales poder realizar estudios. En este trabajo nos centramos en las imágenes de resonancia magnética cardíaca para comparar diversas metodologías de generación de imágenes y explorar las posibilidades que estas nos ofrecen.

Para ello contamos con imágenes anonimizadas de resonancia magnética cardíaca de 397 sujetos, 270 hombres y 127 mujeres, entre los cuales encontramos pacientes de diferentes patologías y sujetos sanos. Estas han sido utilizadas como entrada en diferentes técnicas de generación de imágenes como el Variational Autoencoder (VAE), Convolutional Variational Autoencoder (CVAE) y Generative Adversarial Network (GAN). Además, se ha complementado la generación de las imágenes con la técnica de transferencia de estilo para dotarlas de mayor realismo. Las técnicas que mejores resultados nos han ofrecido han sido CVAE junto a la transferencia de estilo, y gracias a la combinación de las cuales hemos obtenido nuestra herramienta de generación de imágenes de resonancia magnética cardíaca.

Además, para facilitar la visualización de los resultados obtenidos con la transferencia de estilo y explorar la relación entre los diferentes parámetros y las imágenes resultantes, se ha desarrollado una herramienta de visualización que permite además entender el funcionamiento de la transferencia de estilo de forma sencilla.

# Abstract

Artificial intelligence techniques have shown great potential in the area of cardiology. Due to the difficulties involved in working with images of real patients, more and more techniques are being used to generate images with the highest degree of similarity to the real ones, from which to perform studies. In this work we focus on cardiac magnetic resonance imaging to compare different imaging methodologies and explore the possibilities they offer.

For this purpose, we have anonymized cardiac magnetic resonance images of 397 subjects, 270 men and 127 women, including patients with different pathologies and healthy subjects. These images have been used as input in different image generation techniques such as Variational Autoencoder (VAe), Convolutional Variational Autoencoder (CVAe) and Generative Adversarial Network (GAN). In addition, the generation of the images has been complemented with the style transfer technique to make them more realistic. The techniques that have given us the best results have been CVAe together with style transfer, and thanks to the combination of which we have obtained our cardiac magnetic resonance image generation tool.

In addition, to facilitate the visualization of the results obtained with style transfer and to explore the relationship between the different parameters and the resulting images, a visualization tool has been developed that also makes it possible to understand how style transfer works in a simple way.

# Índice

<b>1. Objetivos</b>	<b>1</b>
<b>2. Introducción</b>	<b>2</b>
2.1. Imagen cardíaca por resonancia magnética . . . . .	3
2.1.1. Resonancia magnética . . . . .	3
2.1.2. Anatomía cardíaca . . . . .	3
2.1.3. Enfermedades relacionadas . . . . .	3
2.1.4. Secuencia de imagen cardíaca en RM . . . . .	4
2.2. Redes neuronales artificiales . . . . .	4
2.2.1. Perceptrón multicapa . . . . .	5
2.2.2. Retropropagación . . . . .	5
2.2.3. Redes neuronales convolucionales . . . . .	6
2.2.4. Autoencoders . . . . .	7
2.2.5. Modelos generativos . . . . .	7
2.2.5.1. Variational Autoencoder y Convolutional Variational Autoencoder . . . . .	8
2.2.5.2. Redes generativas adversarias . . . . .	8
2.2.5.3. Transferencia de estilo . . . . .	9
<b>3. Materiales</b>	<b>12</b>
3.1. Hardware . . . . .	12
3.2. Software . . . . .	12
3.3. Dataset . . . . .	12
<b>4. Métodos</b>	<b>14</b>
4.1. Data Augmentation . . . . .	14
4.2. Variational Autoencoder . . . . .	15

4.3. Convolutional Variational Autoencoder . . . . .	15
4.4. Generative Adversarial Network . . . . .	16
4.5. Style Transfer . . . . .	19
4.5.1. Dashboard . . . . .	20
<b>5. Resultados y Discusión</b>	<b>21</b>
5.1. Variational Autoencoder . . . . .	21
5.2. Convolutional Variational Autoencoder . . . . .	21
5.3. Generative Adversarial Network . . . . .	24
5.4. Style Transfer . . . . .	25
5.4.1. Resultados con VGG19 . . . . .	25
5.4.2. Resultados con el <i>encoder</i> del CVAe . . . . .	27
5.4.3. Dashboard . . . . .	27
<b>6. Conclusiones</b>	<b>29</b>
6.1. Líneas futuras . . . . .	29
<b>7. Publicaciones derivadas de este TFM</b>	<b>30</b>
<b>Referencias</b>	<b>31</b>

# 1. Objetivos

En la investigación relacionada con la salud, en muchos casos se utilizan datos de pacientes reales. Dichos pacientes deben dar su consentimiento por escrito de que se utilicen sus datos para fines de investigación. Además, estos datos contienen información sensible de los pacientes, y que está protegida por la ley de protección de datos. Por ello, es necesario que el hospital realice un proceso de anonimización de los datos para proteger la información sensible de los pacientes y evitar que se filtre. Dicho proceso no suele ser una prioridad.

A este complejo proceso de obtención de datos para investigación se le suma que en algunos casos, como en la resonancia magnética, el procedimiento para obtener imágenes es lento y caro. A su vez, también lo es la clasificación, es decir, el diagnóstico de dichas imágenes.

Es por todo ello que existe dificultad en realizar investigación de calidad relacionada con la resonancia magnética debido a la complejidad para obtener imágenes. Esta dificultad aumenta cuando la investigación está relacionada con técnicas de aprendizaje máquina, y sobre todo con técnicas de aprendizaje profundo, en donde se requiere una gran cantidad de datos para poder obtener resultados significativos.

Es necesario por tanto disponer de herramientas que nos permitan aumentar la cantidad de imágenes de las que se dispone para poder realizar investigaciones. En este trabajo por tanto se tiene como objetivo desarrollar una herramienta con la que poder generar imágenes sintéticas con las que poder aumentar el conjunto de datos de una forma rápida, barata y sencilla mediante técnicas de generación de imágenes con aprendizaje profundo.

Para poder cumplir dicho objetivo general, se plantean los siguientes objetivos específicos:

- Explorar diversas metodologías de generación de imágenes, desarrollando un primer generador e ir mejorándolo iterativamente.
- Desarrollar herramientas complementarias que permitan un mayor grado de realismo en caso de ser necesario.

## 2. Introducción

En el presente trabajo, siguiendo los objetivos específicos planteados, se han explorado las metodologías de generación de imágenes Variational Autoencoder, Convolutional Variational Autoencoder y Generative Adversarial Network, de las cuales, como veremos más adelante, la segunda ha sido la que mejor resultados ha dado. También, siguiendo los objetivos específicos planteados, se ha desarrollado una herramienta complementaria para mejorar el realismo de las imágenes generadas mediante la técnica de transferencia de estilo (Style Transfer).

Se ha utilizado la técnica Data Augmentation como parte del preprocesado para aumentar el conjunto de datos, y un Dashboard como herramienta de visualización de las imágenes con transferencia de estilo. Con todo esto es con lo que finalmente hemos desarrollado el generador de imágenes sintéticas de resonancia magnética cardíaca, cuyo esquema podemos ver en la figura 1.

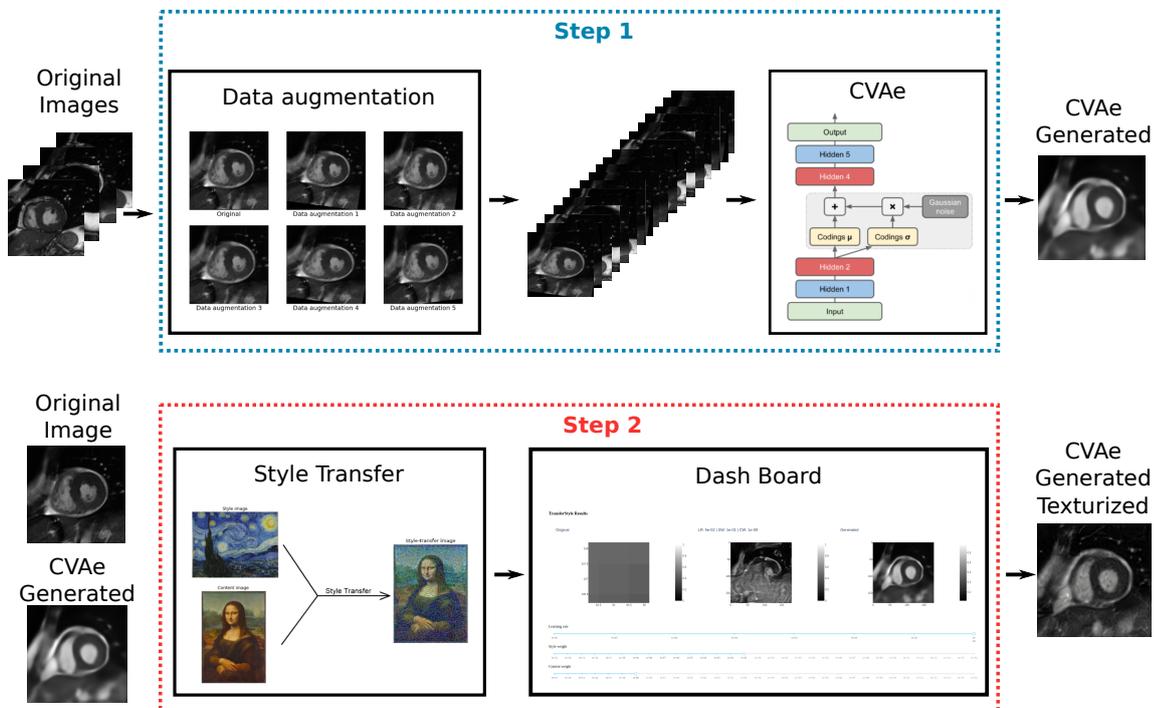


Figura 1: Esquema del generador de imágenes propuesto.

En las siguientes secciones vamos a introducir algunos conceptos necesarios para facilitar la comprensión de éste trabajo.

En primer lugar vamos a describir conceptos relacionados con los datos que hemos utilizado, como son la resonancia magnética, la anatomía y enfermedades del corazón, y la secuencia de imagen cardíaca en resonancia magnética.

Por otro lado, se introducen también conceptos básicos relacionados con el aprendizaje profundo y las redes neuronales, así como de los modelos generativos utilizados.

## **2.1. Imagen cardíaca por resonancia magnética**

### **2.1.1. Resonancia magnética**

La resonancia magnética es un fenómeno físico relacionado con partículas o núcleos atómicos con un número impar de protones y/o neutrones. Estas partículas tienen la propiedad de absorber selectivamente energía de radiofrecuencia cuando son sometidas a un campo magnético.

Las imágenes de resonancia magnética aprovechan esta resonancia de los núcleos de hidrógeno presentes en el cuerpo humano. Concretamente, se utiliza la liberación de la energía absorbida durante la resonancia, a partir de ondas de radiofrecuencia que son recogidas por una antena receptora.

Este fenómeno de liberación es conocido como relajación, y la señal emitida depende de factores propios de los tejidos corporales así como de factores externos tales como la intensidad del campo magnético aplicado. Esto permite obtener no solo imágenes morfológicas, sino también información funcional como puede ser la difusión, o información bioquímica [1].

### **2.1.2. Anatomía cardíaca**

El corazón es el principal órgano en el sistema cardiovascular, encargado de hacer circular la sangre por los vasos sanguíneos para que llegue a todos los órganos y tejidos.

El corazón se compone de cuatro cavidades, que son dos aurículas (izquierda y derecha) y dos ventrículos (izquierdo y derecho). Las aurículas es por donde entra la sangre al corazón, que pasa a través de las válvulas mitral y tricúspide a los ventrículos izquierdo y derecho respectivamente. Los ventrículos son los encargados de expulsar la sangre hacia las arterias para que llegue a todo el cuerpo. Es por ello que los ventrículos están rodeados de una gruesa pared muscular, especialmente el ventrículo izquierdo, que es el que va suministrar la sangre a todo el cuerpo, mientras que el derecho la hace llegar a los pulmones. Podemos ver un esquema de las partes del corazón en la figura 2.

El funcionamiento normal del corazón incluye dos fases diferenciadas en el ciclo cardíaco que son la sístole, correspondiente a la contracción, y la diástole, que se corresponde con la relajación.

### **2.1.3. Enfermedades relacionadas**

En muchos casos, el no correcto funcionamiento del ciclo cardíaco puede causar graves problemas debido a la insuficiencia en el riego sanguíneo, lo que puede afectar negativamente a la salud. En algunos casos existe relación entre el mal funcionamiento del ciclo cardíaco y daños en el tejido cardíaco tales como necrosis (muerte celular), fibrosis (sustitución del tejido por tejido fibroso no funcional) o isquemia (disminución de riego sanguíneo).

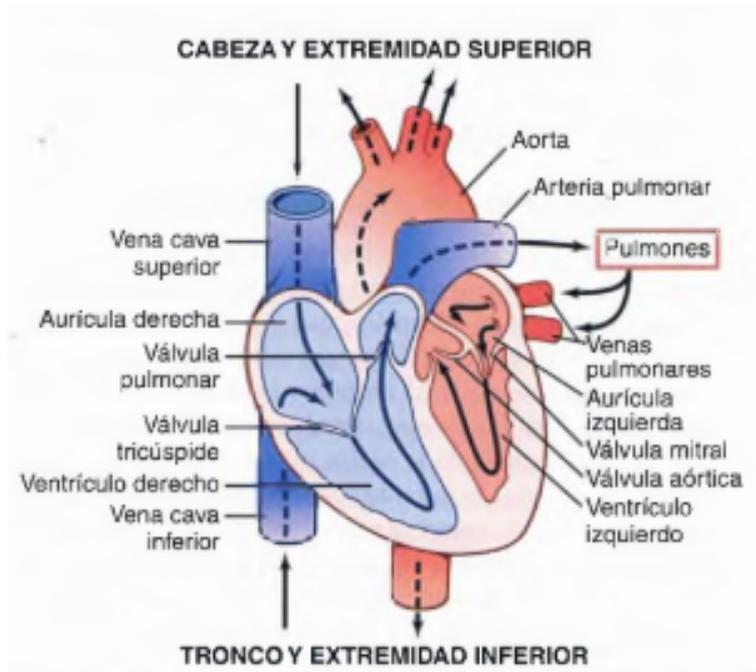


Figura 2: Estructura del corazón y trayecto del flujo sanguíneo a través de las cavidades cardíacas [2].

#### 2.1.4. Secuencia de imagen cardíaca en RM

Para poder evaluar correctamente el ciclo cardíaco es necesario poder observarlo en un ciclo completo. Por ello, es necesario obtener imágenes de todo el corazón de toda la secuencia temporal que comprende el ciclo.

Para poder obtener dichas imágenes, se hace uso de máquinas de resonancia magnética capaces de capturar la información en un corto periodo de tiempo, de forma que se puedan tomar varios cortes dentro de un ciclo cardíaco. Para evitar movimientos torácicos no relacionados con el corazón, tales como la inspiración y la espiración, es necesario que el paciente aguante la respiración durante la toma de imágenes [3].

## 2.2. Redes neuronales artificiales

Se trata de un tipo de modelo computacional que consiste en un conjunto de unidades llamadas neuronas artificiales conectadas entre si. La información de entrada atraviesa la red, donde se va transformando en las diferentes capas de neuronas, hasta producir unos valores de salida.

La utilización de este tipo de modelo computacional constituye el pilar fundamental del aprendizaje profundo, una de las ramas del aprendizaje máquina que se enmarca en el campo de la inteligencia artificial.

### 2.2.1. Perceptrón multicapa

El perceptrón es una de las arquitecturas más simples de redes neuronales artificiales, en donde se toma la información de entrada multiplicada por un conjunto de pesos, se suma, y se le aplica una función de activación. Esto se corresponde con la siguiente ecuación:

$$h = f(\mathbf{x}^T \mathbf{w}) \quad (1)$$

En donde  $f$  es la función de activación,  $\mathbf{x}$  los datos de entrada y  $\mathbf{w}$  los pesos correspondientes a cada una de las posibles entradas. Podemos ver un esquema visual del funcionamiento en la figura 3.

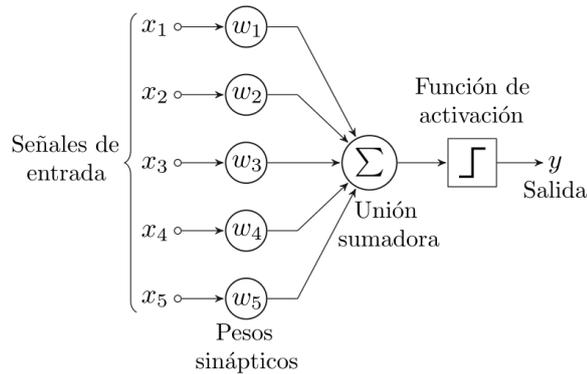


Figura 3: Diagrama de un perceptrón con cinco señales de entrada. Autoría de Alejandro Cartas.

Llamamos perceptrón multicapa a la agrupación de dos o más capas de perceptrones, conectados los perceptrones con las salidas de los perceptrones de la capa anterior, de forma que la salida de un perceptrón es una de las entradas de los perceptrones de la capa siguiente. Esto permite resolver problemas linealmente no separables, una de las principales limitaciones del perceptrón. En la figura 4 podemos observar la conexión entre las diferentes capas de un perceptrón multicapa. Comúnmente se le llama capa de entrada a los datos entrantes, capa de salida a la última capa de perceptrones que va a generar los datos de salida, y al resto de capas se les llama capas ocultas [4].

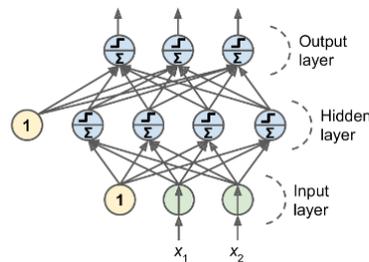


Figura 4: Esquema del perceptrón multicapa [4].

### 2.2.2. Retropropagación

Como hemos visto, las dos partes fundamentales del perceptrón multicapa son los pesos y la función de activación. Mientras que la función de activación es una función fija,

generalmente no lineal, los pesos si van a ir variando a lo largo del entrenamiento de la red. De hecho, va a ser mediante esta variación en los pesos que la red aprenda a resolver el problema para el que se la está entrenando.

Esta variación de los pesos se hace con la intención de minimizar un error dado por una función de coste definida. Por tanto, estamos ante un problema de optimización, en donde debemos encontrar el conjunto de pesos con el que minimicemos la función de error. Esto es lo que entendemos como aprendizaje de la red.

Para realizar este aprendizaje, se utiliza la retropropagación. Se trata de la utilización del descenso por gradiente para encontrar los pesos óptimos. Para ello se le da dos pasadas a la red, una hacia adelante y otra hacia atrás, encontrando el gradiente del error de la red con respecto a cada uno de los parámetros del modelo. Es decir, se calcula cómo debe variar cada peso para reducir el error, para posteriormente aplicar un paso normal del descenso por gradiente.

Así pues, la actualización de los pesos sigue la siguiente fórmula:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (2)$$

Siendo  $\eta$  la tasa de aprendizaje y  $\frac{\partial E}{\partial w_{ij}}$  la derivada parcial del error respecto al parámetro  $w_{ij}$  a actualizar.

Debido al uso de derivadas parciales en la retropropagación, es importante que las operaciones y funciones de activación utilizadas sean diferenciables.

### 2.2.3. Redes neuronales convolucionales

En lo relacionado con las imágenes, el número necesario de parámetros escala a números inmanejables, ya que una primera capa de 1000 perceptrones necesitaría un total de 10 millones de parámetros para una imagen de  $100 \times 100$  píxeles.

Ante este problema, y siguiendo el funcionamiento de las neuronas en la visión y el neocognitrón [5], se utilizan capas de convoluciones, de forma que las neuronas en la primera capa convolucional no están conectadas con todos los píxeles, sino sólo con aquellos de su campo receptivo. De la misma forma, una neurona de una segunda capa convolucional no estaría conectada con las salidas de todas las neuronas de la primera capa, sino a aquellas neuronas localizadas en un pequeño rectángulo, y así sucesivamente. Esta arquitectura permite a la red extraer características de bajo nivel en las primeras capas e ir combinándolas para formar características de alto nivel en las siguientes capas. Podemos ver un ejemplo del funcionamiento jerárquico en la figura 5.

Una de las primeras aplicaciones fue el reconocimiento de caracteres manuscritos, mediante la famosa arquitectura *LeNet-5* [6], que abrió el camino a un gran desarrollo en el campo de la visión por ordenador.

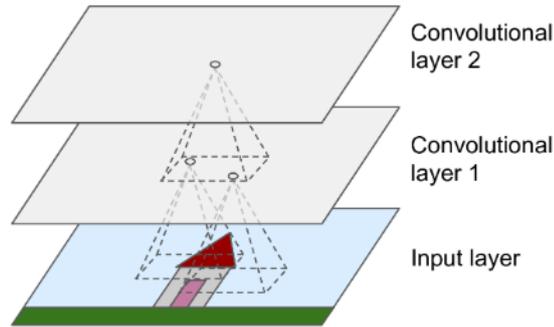


Figura 5: Campo receptivo de las neuronas en las capas convolucionales [4].

#### 2.2.4. Autoencoders

Un autoencoder es un tipo de arquitectura de red neuronal en donde se diferencian dos módulos separados. Estos módulos son el codificador o *encoder* y el decodificador o *decoder*. Generalmente tienen una arquitectura espejo, en donde el encoder se encarga de codificar la información entrante en un espacio latente de menor dimensión que los datos de entrada, y el decoder debe, a partir de la información codificada en dicho espacio latente, ser capaz de recrear los datos originales. En la figura 6 podemos ver una arquitectura típica de autoencoder.

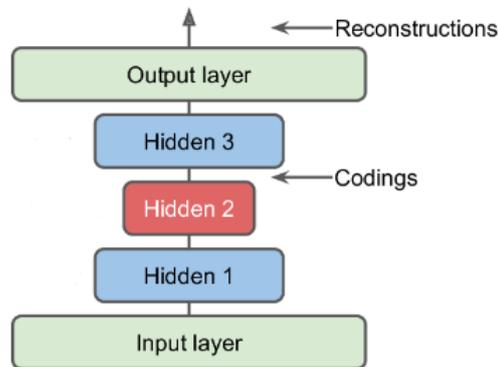


Figura 6: Esquema de la arquitectura básica en los autoencoders [4].

Esto es útil en muchos tipos de tareas, como en la reducción de dimensionalidad o en técnicas de *clustering*. También es común ver su utilización en aprendizaje auto supervisado, en donde se utiliza el autoencoder para que la red aprenda las características internas de los datos para posteriormente utilizar lo aprendido para resolver un problema.

#### 2.2.5. Modelos generativos

Los modelos generativos son un tipo de modelos de redes neuronales utilizados para la generación de datos. Entre los más destacados tenemos los Variational Autoencoders y su variante convolucional, los Generative Adversarial Networks y la transferencia de estilo.

### 2.2.5.1 Variational Autoencoder y Convolutional Variational Autoencoder

Aunque la utilización de los autoencoders data de los años noventa [7], trabajos más recientes han vinculado los autoencoders con modelos de variable latente, permitiendo así el uso de los autoencoders como modelos generativos.

El Variational Autoencoder (VAe) es un tipo especial de autoencoder, el cual puede ser interpretado como un autoencoder probabilístico. Es decir, sus salidas son parcialmente determinadas por un componente estocástico, incluso en la inferencia. Deben su nombre a que hacen uso de la inferencia Bayesiana variacional.

La principal diferencia con los autoencoders normales es que el encoder no produce directamente una codificación de los datos en el espacio latente, sino que produce una codificación media  $\mu$  y una desviación estándar  $\sigma$ , de forma que la codificación final se toma aleatoriamente de una distribución normal de media  $\mu$  y desviación estándar  $\sigma$ .

De esta forma, la salida no pertenece a la codificación de la entrada, sino a la codificación de unos datos que están muy próximos en el espacio latente a los datos de entrada, y por tanto se espera que sean muy similares a estos. Podemos ver un esquema del funcionamiento del VAe en la figura 7 [4, 8, 9].

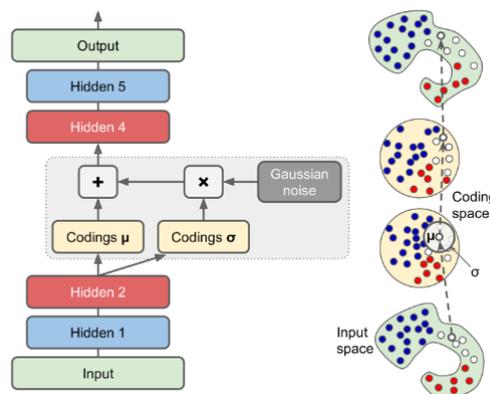


Figura 7: Arquitectura y funcionamiento de un VAe [4].

En el campo de la generación de imágenes, se modifica la arquitectura típica del VAe para añadir capas convolucionales, aunque manteniendo la idea de arquitectura espejo y muestreo a partir de distribución normal. A este tipo de VAe se le denomina Convolutional Variational Autoencoder (CVAe).

### 2.2.5.2 Redes generativas adversarias

En 2014, Ian Goodfellow et al. [10] proponen una nueva metodología para los modelos generativos. Se trata de las redes generativas adversarias (GAN por sus siglas en inglés), con una idea sencilla pero potente que generó mucho entusiasmo entre la comunidad. Las GANs, como su nombre indica, consiste en enfrentar a dos redes distintas, un generador y un discriminador, de forma que compitan entre ellas.

De esta forma, el generador tomará un vector de ruido a partir del cual deberá generar

una salida. Este vector de ruido es análogo a la codificación en el espacio latente que hemos visto en los VAe, y a su vez el generador tiene la misma función que el decodificador. Son análogos.

Por otro lado, el discriminador toma datos generados por el generador y datos reales, y debe aprender a diferenciar los datos reales de los generados.

Podemos ver en la figura 8 un esquema del funcionamiento de las GAN, donde el generador va a intentar “engañar” al discriminador, tratando de hacer datos tan realistas que el discriminador no pueda diferenciarlos, mientras que el discriminador va a intentar evitar que el generador le “engañe”.

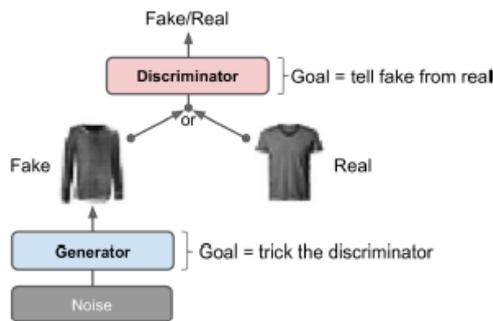


Figura 8: Esquema del funcionamiento de las GAN [4].

El entrenamiento de las GAN, al contrario del resto de arquitecturas, tiene dos fases diferenciadas debido a que tenemos dos redes diferentes, que hay que ir entrenando de forma alterna:

- En la primera fase, generamos datos para, junto a una muestra del conjunto de datos reales, entrenar el discriminador. Los datos reales tendrán como etiqueta 1, mientras que los generados 0. En esta fase tan solo se actualizan los pesos del discriminador.
- En la segunda fase generamos nuevamente imágenes con el generador, con la diferencia de que en este caso al discriminador tan solo le pasaremos dichos datos generados, ahora como 1. Es decir, vamos a entrenar al modelo para que produzca imágenes que el discriminador clasificaría como reales. En este punto se congelan los pesos del discriminador, modificando solo los del generador.

Dentro de las GAN existe una variación llamada Deep Convolutional GANs (DCGAN) en donde se aplica la misma metodología utilizando capas convolucionales, lo que las hace especialmente interesantes para la generación de imágenes.

### 2.2.5.3 Transferencia de estilo

La transferencia de estilo es una técnica de síntesis de texturas que hace uso de CNN para la extracción y transferencia de características de la imagen. En esta técnica, se

utiliza la capacidad de extraer características de la CNN utilizada para, mediante la retropropagación, modificar la imagen tal que las características de bajo nivel sean similares a las de la imagen de la cual se quiere obtener el estilo (las texturas), y las características de alto nivel sigan siendo lo más similares posible a la imagen de la cual se quiere el contenido.

En el artículo original se utiliza la arquitectura VGG19, entrenada con el conjunto de datos de ImageNet para el reconocimiento de objetos. Esta red por tanto ha sido entrenada con miles de imágenes y ha aprendido a extraer las características de alto y bajo nivel de las imágenes.

La metodología de trabajo pues es pues obtener los mapas de características de bajo nivel de la imagen de estilo, los de alto nivel de la imagen de contenido, e ir aplicando retropropagación a la imagen de contenido para modificar sus píxeles. La función de coste a utilizar va a comparar los mapas de características de nuestra imagen con los obtenidos previamente, estando por un lado la función de coste de los de estilo y por otro lado la función de coste de contenido, siendo la función de coste final la suma ponderada de estas dos.

Tendremos por tanto dos hiperparámetros importantes en la transferencia de estilo, que son el peso que le damos al estilo y el que le damos al contenido, que van a afectar al resultado final.

Un ejemplo sencillo es el caso en el que aplicamos texturas de pinturas famosas, como la noche estrellada, a una La Mona Lisa, de modo que obtenemos una imagen de dicha pintura como si la hubiese pintado Van Gogh, tal como podemos ver en la figura 9 [11].

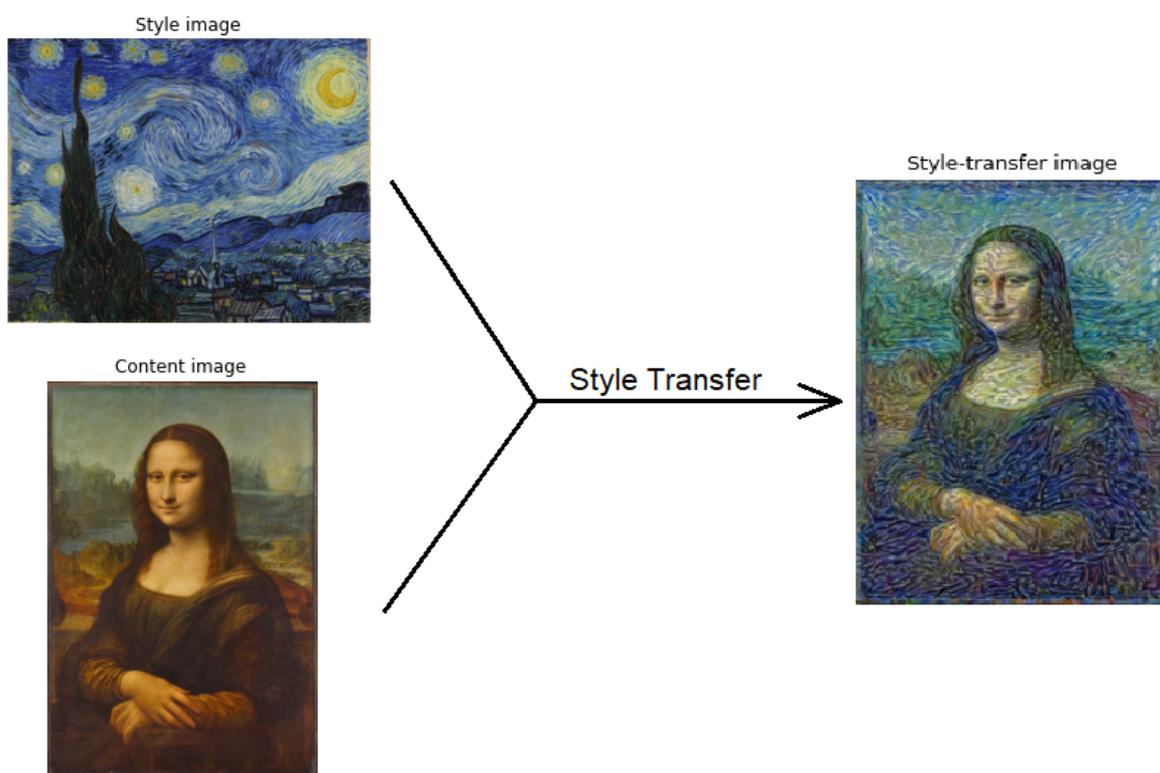


Figura 9: Transferencia del estilo de La noche estrellada de Van Gogh a La Mona Lisa.

## 3. Materiales

### 3.1. Hardware

Para la realización de este trabajo se ha utilizado un ordenador con una CPU Intel i9-9900KF y una GPU Nvidia GeForce RTX 2080 Ti de 11GB.

### 3.2. Software

El software utilizado ha sido Windows como sistema operativo. Los experimentos se han realizado utilizando el lenguaje de programación Python junto con diversas librerías como Numpy, Pandas, Matplotlib, Dash y Plotly. Además, los frameworks Tensorflow y Keras también se han utilizado para realizar las diferentes redes neuronales utilizadas.

El código utilizado se puede encontrar en el repositorio de github <https://github.com/jmjaenlorites/TFM>, donde podemos encontrar los diferentes *Jupyter notebooks* utilizados.

### 3.3. Dataset

Para la realización de este trabajo, se ha contado con un conjunto de datos anonimizado dado por el Hospital Clínico Universitario de Valencia, con el consentimiento escrito de los pacientes. Este conjunto de datos consistía en 397 imágenes de resonancia magnética de eje corto conteniendo tanto el ventrículo izquierdo como el derecho del corazón. El dataset comprende un total de 397 pacientes (270 hombres y 127 mujeres), con una edad de  $64.53 \pm 12.35$  años ( $63.27 \pm 11.98$  años los hombres y  $67.45 \pm 12.75$  las mujeres). El conjunto de pacientes fue diagnosticado con diferentes enfermedades cardíacas entre las que se encuentran fibrosis, isquemia, necrosis, afectación funcional del ventrículo izquierdo y/o del derecho, así como también hay pacientes sanos. Para los experimentos se han utilizado las imágenes correspondientes a la sístole.

La toma de imágenes se ha realizado en “breath-hold” (el paciente aguanta la respiración para evitar movimientos no cardíacos) con un escáner de resonancia magnética de 1.5 T (Sonata Magnetom Siemens, Erlangen, Germany), con un ángulo de giro de  $58^\circ$ , tiempo de repetición de 52.92 ms y tiempo de eco de 1.25 ms.

La resolución en el plano varió en todos los casos, oscilando entre  $0.57 \times 0.57$  mm<sup>2</sup> y  $1.09 \times 1.09$  mm<sup>2</sup>. El grosor del corte y el espacio entre cortes fue constante en todos los casos, 7 mm y 3 mm respectivamente. Los tamaños de las imágenes resultantes variaron de  $144 \times 144$  a  $256 \times 256$  y el número de cortes osciló entre 8 y 14. Todas las imágenes se remuestrearon mediante interpolación bilineal a una resolución espacial constante en el plano de 2 mm<sup>2</sup> con un tamaño de imagen de  $180 \times 180$ . El eje z se dejó intacto en el proceso de reestructuración. Además, las pilas 3D se normalizaron para que los valores de los píxeles fueran de 0 a 1 utilizando la normalización min-max.

Podemos ver algunos ejemplos de imágenes de resonancia magnética en la figura 10.

### Ejemplos de imágenes de resonancia magnética cardíaca

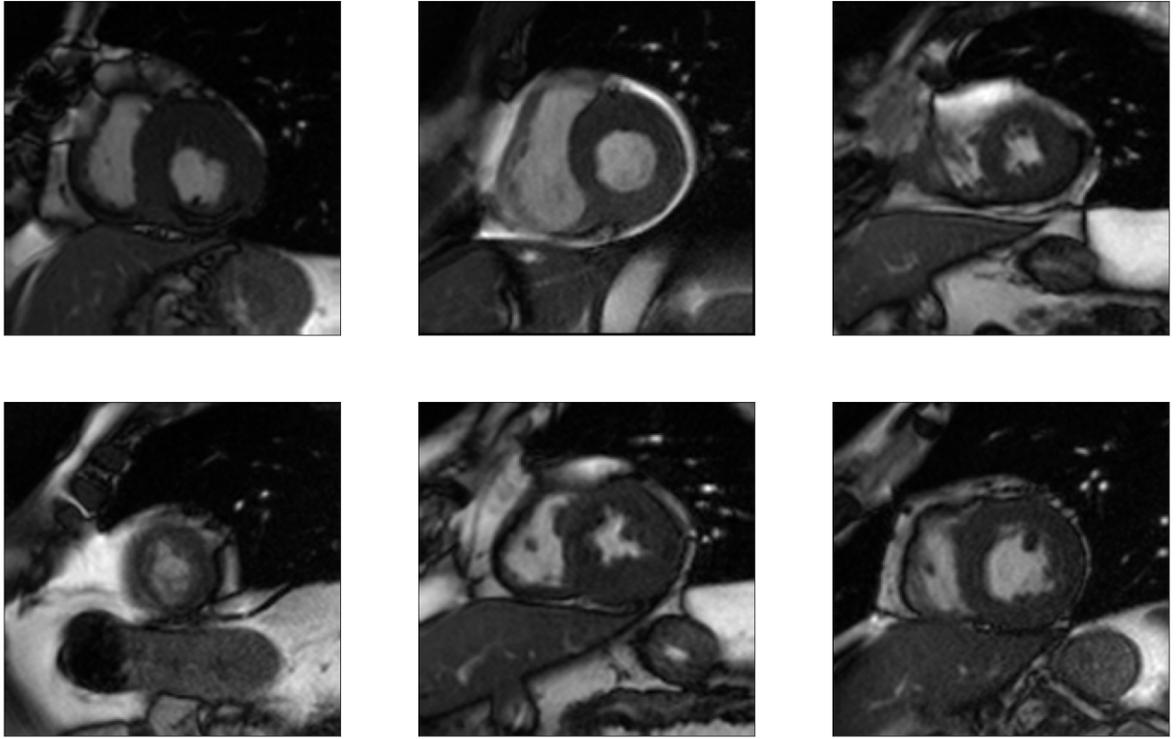


Figura 10: Ejemplos de imágenes de resonancia magnética cardíaca.

## 4. Métodos

### 4.1. Data Augmentation

Tras un breve análisis exploratorio de los datos en donde se confirman las características mencionadas en el apartado del Dataset en Materiales, se procede a aplicar la técnica conocida como data augmentation. Esta técnica se utiliza debido al reducido tamaño de nuestro conjunto de datos.

Data augmentation es una técnica utilizada para aumentar el tamaño del conjunto de imágenes mediante la duplicación de éstas, añadiéndoles ligeras modificaciones. Estas modificaciones pueden ser rotaciones, zoom, *cropping* (cortar una parte), voltear la imagen, etc. Con esto, conseguimos tener nuevas imágenes a partir de las originales, de forma que el modelo a entrenar pueda aprender las características intrínsecas de las imágenes cuando éstas son modificadas ligeramente, añadiendo variabilidad a los datos.

Esto se realiza haciendo uso de Tensorflow, que tiene funciones específicas para ello. Concretamente se aumentará el dataset con rotaciones dentro de un rango de  $10^\circ$  y con zoom dentro de un rango de 0.1, generando un total de 3000 imágenes adicionales. Podemos ver un ejemplo del resultado de aplicar Data Augmentation en la figura 11.

Se utilizará un 80 % de estas imágenes para entrenar, siendo el resto para validación.

#### Ejemplos de Data Augmentation

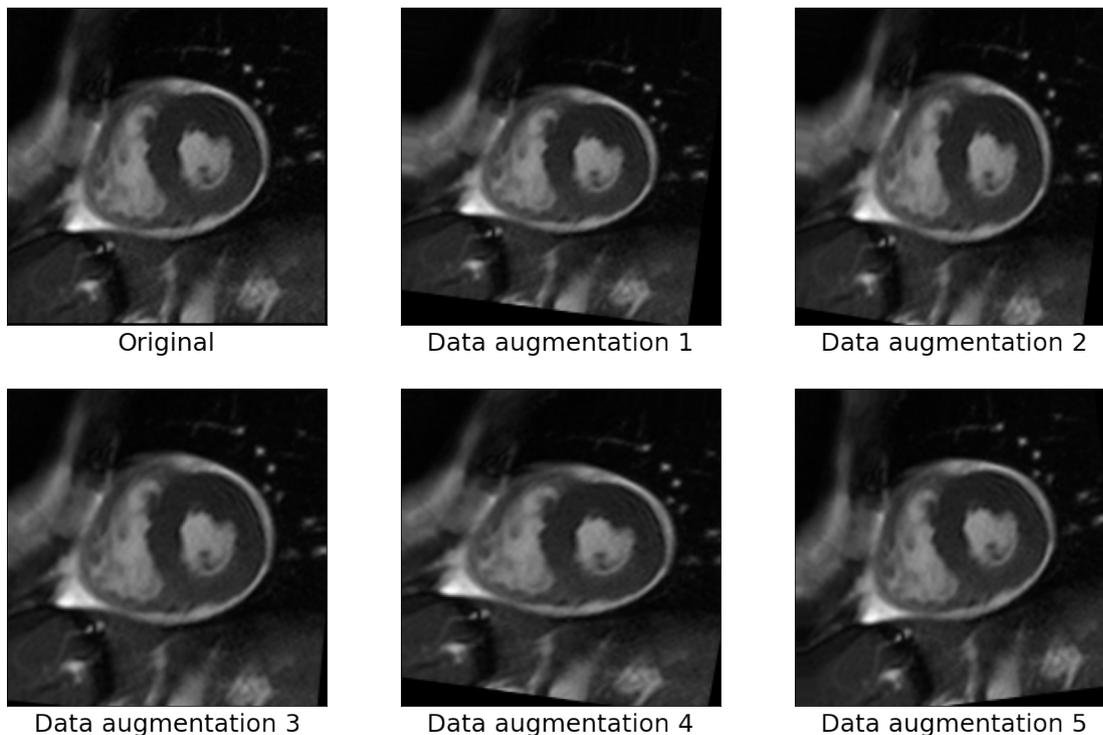


Figura 11: Imagen original junto a obtenidas mediante data augmentation.

## 4.2. Variational Autoencoder

Una vez hemos preparado el dataset con el que trabajar, procedemos con el VAe como primera técnica de generación de imágenes a explorar.

Se han probado diferentes configuraciones hasta llegar a la arquitectura que podemos observar en la figura 12. También se han probado diferentes hiperparámetros, como el tamaño de lote (de 16, 32, 64, 128 y 256), número de épocas (de 10, 50, 100, 250 y 500) y optimizadores (RMSprop, SGD).

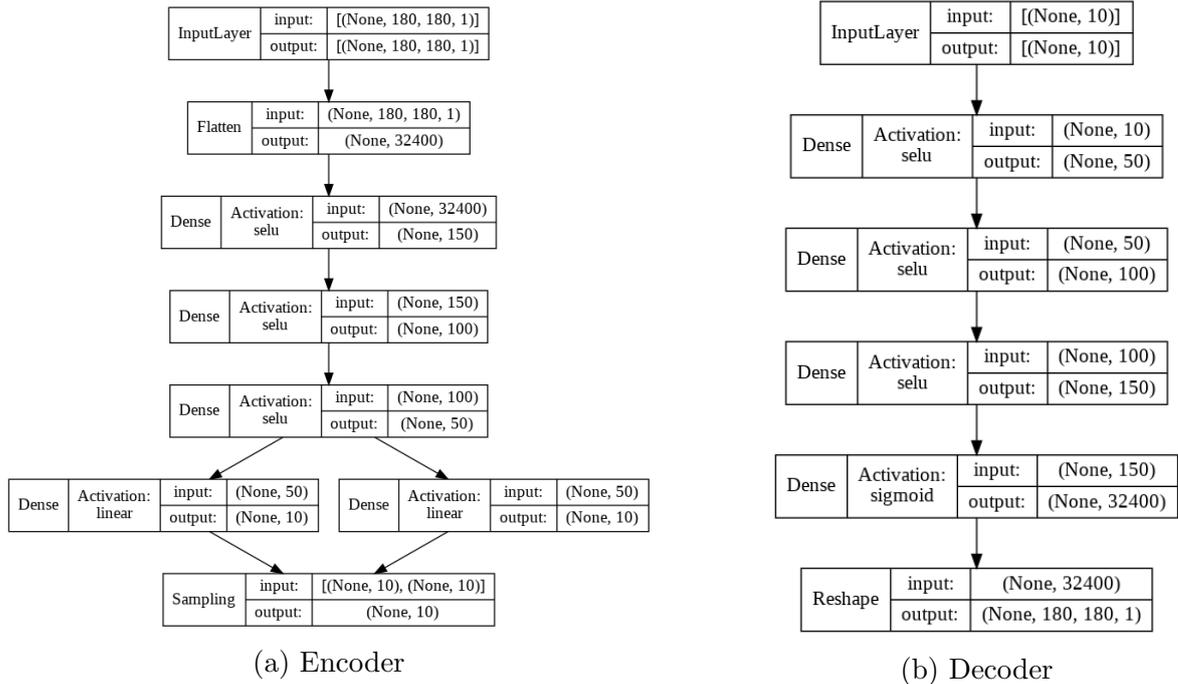


Figura 12: Arquitectura del VAe

Es con esta arquitectura, entrenando el modelo con “binary crossentropy” como función de coste y RMSprop como optimizador, con 500 épocas con un tamaño de lote (batch size) de 256 es con lo que hemos obtenido nuestro primer generador de imágenes, y por tanto un primer modelo a superar. Podemos ver las imágenes generadas con este modelo en la sección 5.1.

## 4.3. Convolutional Variational Autoencoder

El siguiente paso en este trabajo ha sido proceder a aumentar la complejidad de la arquitectura del modelo, aumentando el número de capas y parámetros, e incluyendo las capas convolucionales, empezando a trabajar así con los CVAe.

Se empieza con una arquitectura sencilla, similar a la arquitectura espejo utilizada con el VAe, pero en este caso utilizando capas convolucionales y capas convolucionales transpuestas, junto a capas densas para codificar y decodificar el vector en el espacio latente.

No obstante, los resultados obtenidos son peores que los obtenidos con el VAe (resultados en sección 5.2, figura 17), donde se generan unos extraños artefactos, y con dificultades para llegar a identificar partes de la imagen.

Tras varias investigaciones y pruebas, llegamos a la conclusión que la causa de los artefactos era debido a las capas de convolución transpuesta, y se sustituyen por capas de upsampling junto a capas convolucionales.

A su vez, continuamos realizando modificaciones en la arquitectura, aumentando en gran medida su tamaño respecto al VAe, con una arquitectura con más de un millón de parámetros (1 338 049). Además de la arquitectura espejo, se añaden a continuación más capas convolucionales, con el objetivo de dotar de texturas más realistas a la imagen.

Se sigue ampliando el modelo, aumentando el tamaño del espacio latente de 64 a 512, así como aumentando el número de capas, llegando a tener una arquitectura de más de 4 millones de parámetros. En los diferentes experimentos se utiliza una tasa de aprendizaje en función del número de épocas tal que:

$$lr = (1 - \alpha) \cdot \gamma + \alpha \quad (3)$$

donde  $\alpha$  es el learning rate mínimo ( $5 \cdot 10^{-5}$ ) y  $\gamma$  es el decaimiento del coseno tal que:

$$\gamma = \frac{0,5 \cdot (1 + \cos(\frac{\pi \cdot epoch}{t_0}))}{1 + \frac{epoch}{1500}} \quad (4)$$

Siendo  $t_0$  el periodo del coseno (400 épocas).

El modelo resultante ha sido con la arquitectura de la figura 13 entrenado nuevamente con “binary crossentropy” como función de coste, ahora con Adam como optimizador y una tasa de aprendizaje inicial de 0.0001, con 10 000 épocas y con un batch size de 64 (por razones de capacidad de cómputo). Podemos ver los resultados de este modelo en la sección 5.2.

#### 4.4. Generative Adversarial Network

Alternativamente, se ha experimentado también con las GAN. Concretamente, se ha experimentado con las DCGAN.

La arquitectura utilizada está compuesta por capas de convolución transpuesta en el generador, mientras que tenemos capas convolucionales en el discriminador. Se ha utilizado la “binary crossentropy” nuevamente como función de coste tanto para el generador como para el discriminador. También el optimizador ha sido Adam, con una tasa de aprendizaje de 1e-4 para ambos. Podemos ver la arquitectura en mayor detalle en la figura 14.

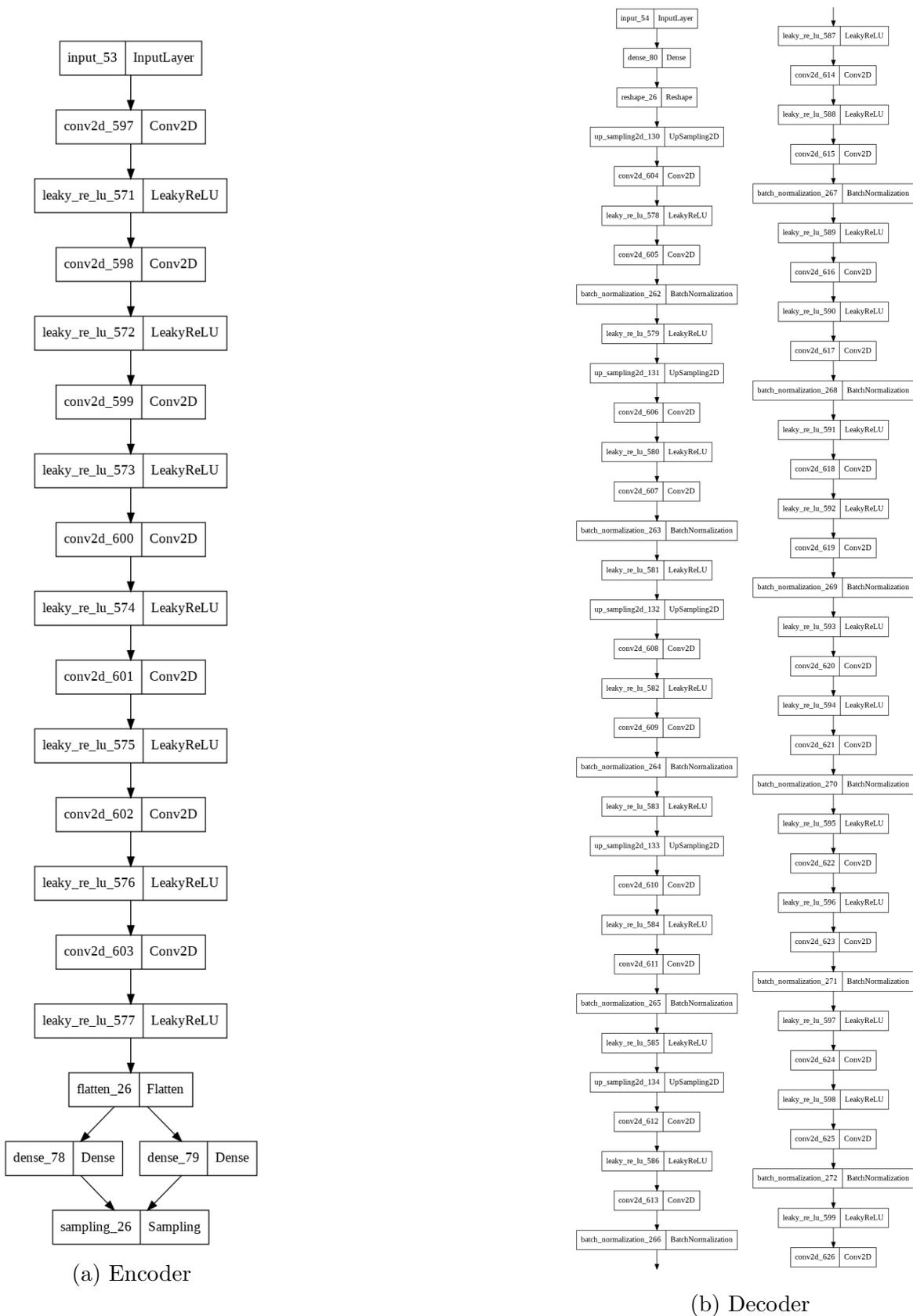
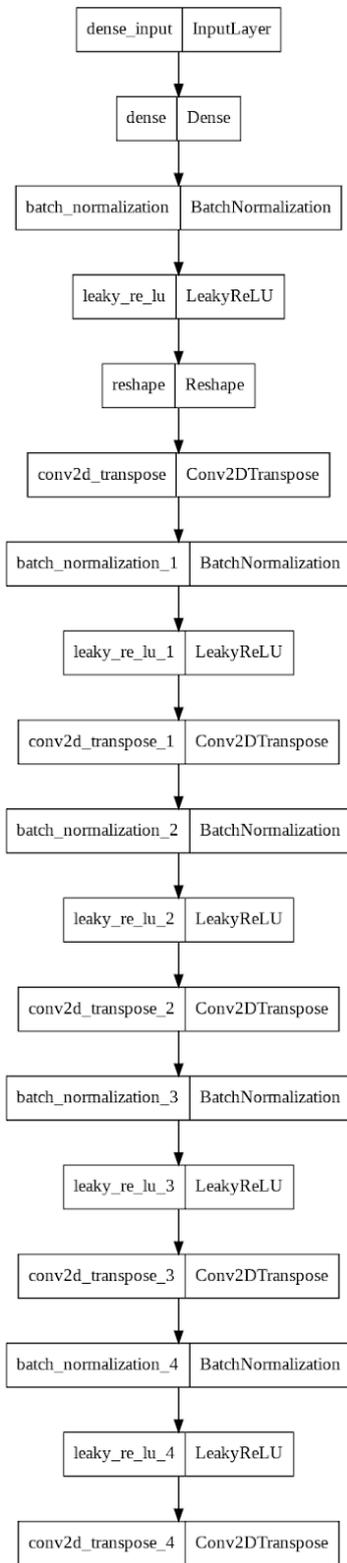
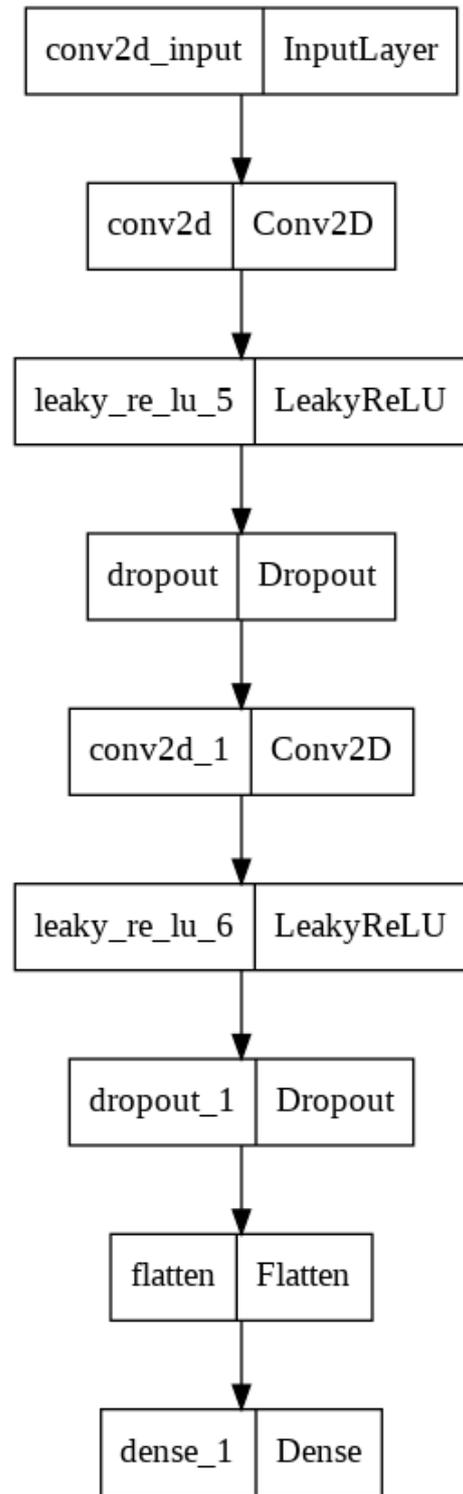


Figura 13: Arquitectura del CVAe



(a) Generador



(b) Discriminador

Figura 14: Arquitectura de la GAN

Se han realizado algunos experimentos con esta arquitectura, aunque sin poder profundizar demasiado debido a la gran complejidad que presentan estas redes. Basta con observar que Skandarani, Jodoin y Lalande [12] mencionan que tuvieron que realizar más de 1500 experimentos hasta poder conseguir resultados notables con este tipo de arquitecturas para darse cuenta de la dificultad que presentan.

## 4.5. Style Transfer

Debido a la falta de detalle en las imágenes generadas (resultados de CVAe en sección 5.2), es decir la falta de texturas, se decide utilizar las técnicas de Style Transfer. De esta forma podremos transferir las texturas de imágenes reales de resonancia magnética cardíaca a imágenes generadas mediante nuestro CVAe. Juntando las dos arquitecturas es con lo que conseguiremos finalmente nuestro generador.

Para la construcción de la red que efectúe el Style Transfer, nos basaremos en las capas internas de la red ya entrenada VGG19. Concretamente utilizaremos las capas “conv1” de los cinco bloques para el estilo, y la capa “conv2” del quinto bloque para el contenido.

Para el entrenamiento hemos utilizado el optimizador Adam, siendo la función de coste el error cuadrático medio entre los mapas de características obtenidos por las capas mencionadas previamente con la imagen de estilo o contenido, y los obtenidos con la imagen que estamos transformando. De esta forma transformamos la imagen minimizando la diferencia entre los mapas de características obtenidos por las capas internas de la red y los obtenidos de la imagen de la que queremos extraer el estilo, así como minimizando la diferencia entre los mapas de características obtenidos para la imagen de contenido y la resultante [11].

Dado que la red que utilizamos esperamos que sea capaz de extraer correctamente la información contenida en la imagen para que podamos comparar correctamente los mapas de características, se prueba también a utilizar las capas internas del propio encoder del CVAe. Resulta también interesante comparar los resultados con los de la VGG19 debido a que esta red está entrenada con imágenes cotidianas, no con imagen médica, por lo que el rendimiento puede verse afectado.

Se realiza un conjunto de experimentos para la combinación de diversos valores de peso de estilo, peso de contenido y tasa de aprendizaje, entrenando con 1000 épocas en cada caso. Esto se repite para la utilización de las capas de los dos modelos mencionados.

Los valores de peso de estilo y de contenido son (1e-15, 1e-14, 1e-13, ..., 1e14, 1e15, 1e16), mientras que los de la tasa de acierto son (0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001).

### 4.5.1. Dashboard

Debido a la gran cantidad de experimentos realizados, y la dificultad de analizar correctamente los resultados, se decide hacer uso de una herramienta de visualización en donde se pueda analizar fácilmente los resultados obtenidos. Para ello se hace uso de framework Dash de Python con el que crear un dashboard. Con este dashboard se pretende poder observar el efecto de la transferencia de estilo en función de los diferentes parámetros utilizados. Tendremos pues la imagen original, la imagen con el estilo transferido y la imagen generada, junto a diferentes deslizadores con los que podremos seleccionar la combinación de parámetros deseada. Tendremos también un menú seleccionable donde indicar el modelo.

Debido a que el volumen de imágenes, si bien es grande, no llega a ser excesivo, con el fin de agilizar el proceso, se realiza la transferencia de estilo a la misma imagen con las diferentes combinaciones de parámetros, serializando el resultado en cada uno de los casos. De esta forma tenemos el resultado almacenado en el disco duro en un formato muy rápido de leer por lo que, cuando indicamos el conjunto de parámetros que queremos ver, no se realiza la transferencia de estilo en sí, sino que se lee y visualiza una imagen generada a la que ya se le ha aplicado la transferencia de estilo.

Esta aplicación se encuentra online en el siguiente enlace\*:

<https://fierce-bayou-68989.herokuapp.com/>

*\*Nota: Puede llevar un tiempo para cargar la primera vez o incluso fallar. Si se diese el caso, reintentar un minuto más tarde.*

## 5. Resultados y Discusión

### 5.1. Variational Autoencoder

Con esta arquitectura se consiguen generar imágenes en donde se llegan a distinguir los dos ventrículos del corazón, así como la estructura anatómica típica de un corte de imagen resonancia cardíaca donde apreciamos la forma de las costillas y el aire del pulmón. Lo podemos ver indicado en la figura 15.

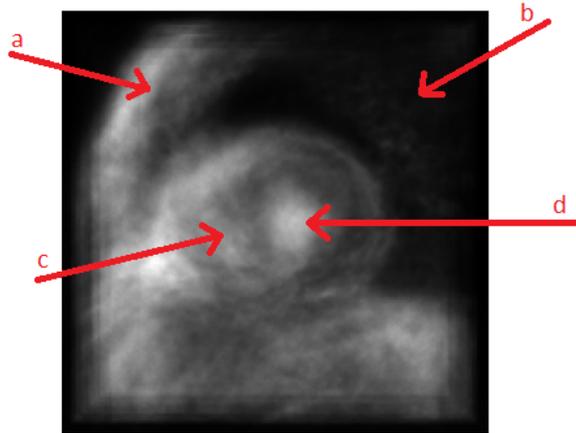


Figura 15: a: costilla; b: pulmón; c y d: ventrículos derecho e izquierdo

No obstante, con el VAe, las imágenes son muy difusas, sin que se llegue a apreciar ningún tipo de detalle, más allá de la estructura anatómica general a grandes rasgos. Podemos ver más ejemplos de imágenes generadas mediante el VAe en la figura 16.

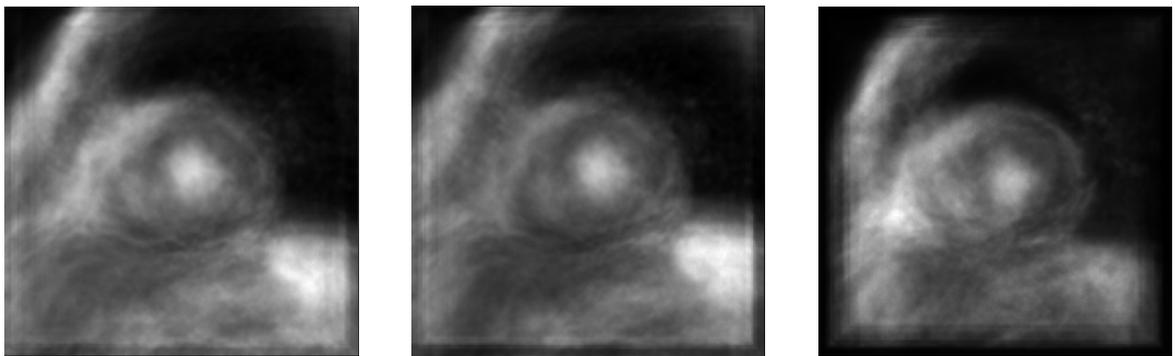


Figura 16: Imágenes generadas por el VAe.

### 5.2. Convolutional Variational Autoencoder

El resultado obtenido con las capas de convolución transpuesta presenta a grandes rasgos cierta estructura general, aunque sin terminar de ser clara. Además, se observa un artefacto que se repite por toda la imagen. La causa de dicho artefacto es debida en parte a la naturaleza de la convolución transpuesta, la cual es una operación muy similar a la convolución, en donde le estamos pidiendo que infiera el muchos más píxeles

de los que se le provee. Es decir, a partir de 9 píxeles debe ser capaz de inferir 36 píxeles, lo que provoca que al buscar la forma más óptima de hacerlo, en el sentido de encontrar los filtros, dado que se compara con un gran conjunto de imágenes, encuentre que los filtros que menor error tienen en la función de coste utilizada provoquen el mencionado patrón. Tenemos un ejemplo de las imágenes generadas con esta arquitectura en la figura 17.

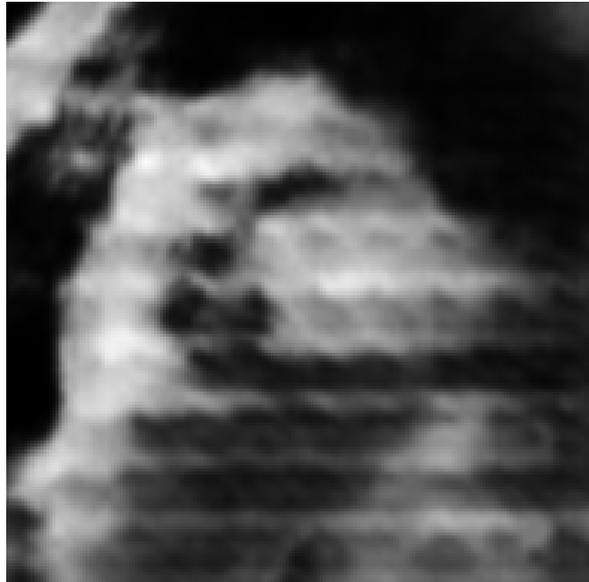


Figura 17: Imagen generada por los primeros CVAe.

Una vez hemos pasado a utilizar capas de upsampling junto a capas convolucionales, vemos desaparecer el molesto patrón. Esto es debido a que ahora el modelo no debe aprender a inferir 36 píxeles a partir de 9, sino que mediante el upsampling se realiza una ampliación de los mapas de características multiplicando el número de píxeles de ancho y de alto por dos. Esta es una operación de interpolación bilineal la cual no tiene parámetros a optimizar. La capa convolucional que le sigue ya no ha de aprender a inferir x4 veces más píxeles de los que se le alimenta, por lo que evitamos la aparición de los mencionados artefactos.

En este punto tenemos imágenes que se parecen a grandes rasgos a imágenes reales. El modelo ha aprendido a capturar la estructura anatómica que se aprecia en la imagen, codificar en el espacio latente, y reconstruir dicha estructura general. Con ello, únicamente con el decodificador podemos crear imágenes que tengan esta estructura anatómica realista a partir de ruido gaussiano, tales como la que podemos apreciar en la figura 18.

No obstante, esta imagen dista de ser realista, ya que aunque su estructura general (baja frecuencia) sí lo sea, carece de detalles (alta frecuencia). Es decir, no tiene la textura típica que presentan las imágenes de resonancia magnética cardíaca.

Hasta donde sabemos, resolver esto no es trivial, ni algo que pueda ser resuelto fácilmente aumentando el número de épocas o el número de capas del modelo. Es además un problema común en tareas relativas a la generación de imágenes. De hecho, en la figura 19 podemos ver como, aún entrenando con 10 000 épocas el modelo, empieza a

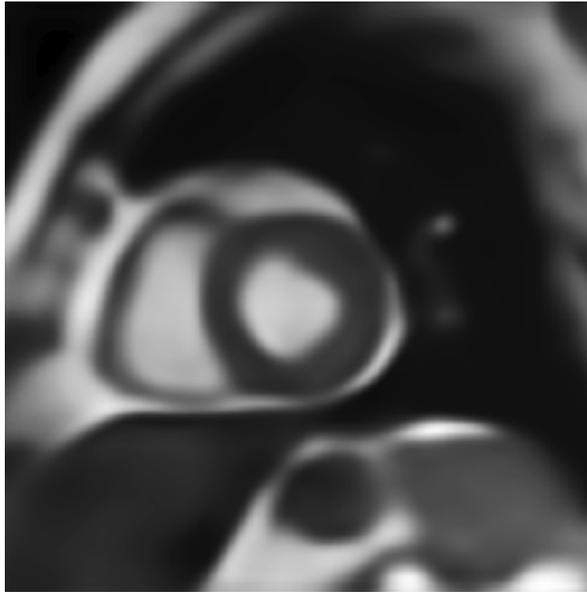
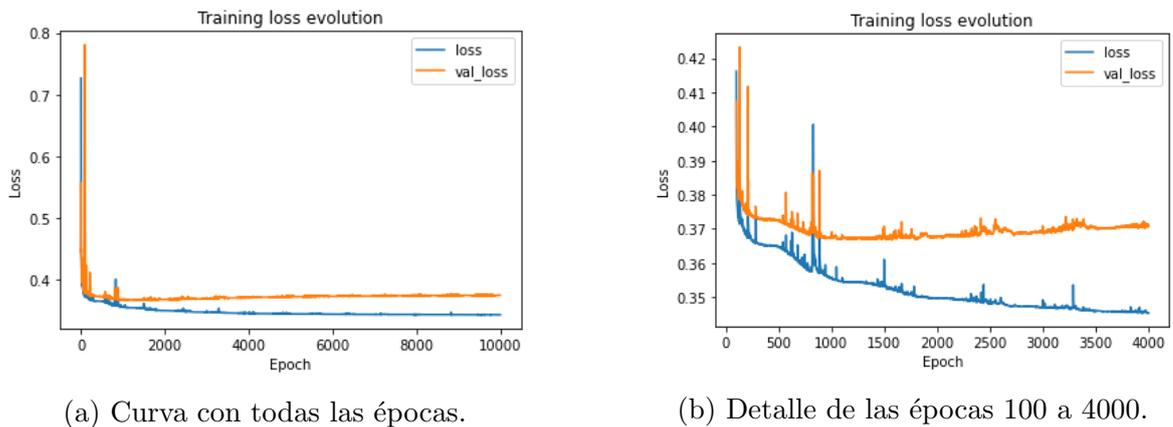


Figura 18: Imagen generada por el CVAe final.



(a) Curva con todas las épocas.

(b) Detalle de las épocas 100 a 4000.

Figura 19: Curva de entrenamiento del CVAe.

haber sobre ajuste a partir de la época 1000 aproximadamente, y dejando de aprender nada a partir de la época 4000.

Podemos observar también en la figura 19b cómo aparecen perturbaciones en las líneas de *loss* y *val\_loss*, las cuales se corresponden al carácter oscilatorio que tiene la tasa de aprendizaje, respondiendo a una función coseno. Es por ello que, cuando esta tasa de aprendizaje está en el punto elevado, tenemos mayores perturbaciones, y cuando está baja, apenas hay perturbaciones, ya que apenas está aprendiendo y el resultado es muy similar a las épocas anteriores.

Cabe recordar nuevamente la forma en la que aprenden las redes neuronales, que es minimizando el error que obtienen a partir de una función de coste. Esto provoca que en muchos casos, ante la dificultad de encontrar los parámetros que nos generarían la imagen completamente real, encuentren la forma de generar imágenes suaves, con poco detalle, de forma que eviten “jugársela” a crear un detalle que en muchos casos conlleva a tener un mayor error en la función de coste.

### 5.3. Generative Adversarial Network

A pesar de no haber profundizado en gran medida en los experimentos relacionados con esta red, sí que se han generado algunas imágenes como las que podemos observar en la figura 20.

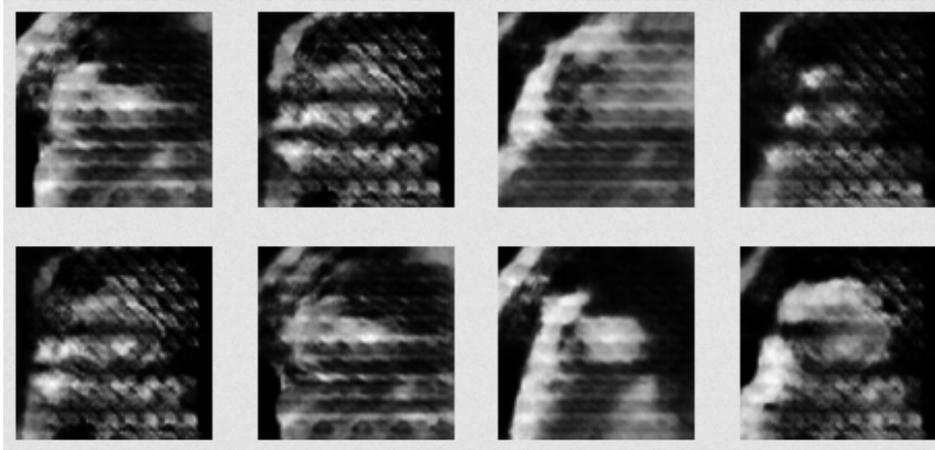


Figura 20: Imágenes generadas por la GAN

Vemos algunos resultados similares a los obtenidos con los primeros experimentos realizados con el CVAe, tanto en forma como en artefactos. Si bien en este caso, los artefactos son más pronunciados y la estructura anatómica menos definida, sí se atisba alguna forma conocida, como las costillas.

Seguramente aplicando algunos de los cambios aplicados en la segunda tanda de experimentos en el CVAe, los resultados de la GAN hubieran resultado más realistas.

Una de las diferencias principales en el entrenamiento entre el CVAe y la GAN, es la función de coste a minimizar, en donde en el CVAe tenemos una única función a minimizar, y fácilmente podemos asociar un menor error en la función de coste con una mejora en la generación. Sin embargo esto no ocurre en las GANs, debido a que tenemos dos módulos claramente diferenciados los cuales entrenamos de forma alterna. Esta competición entre los módulos implica que, en el supuesto caso de que el error de uno de los módulos se reduzca no es necesariamente bueno, sino que puede implicar una ruptura del equilibrio entre los módulos en donde uno de ellos se ha sobrepuesto al otro, impidiendo que aprenda correctamente y por tanto estancándose en un generador pobre.

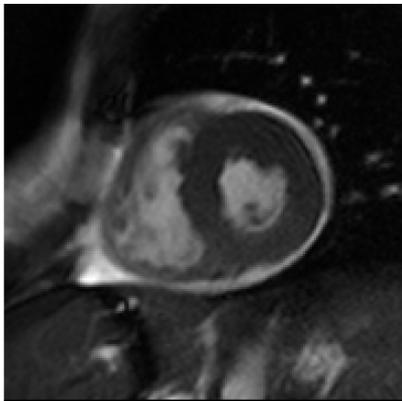
A esto hay que añadirle la posibilidad de que esté yendo por buen camino, generando cada vez mejores imágenes, y en un momento dado se produzca una variación brusca que altere uno de los módulos, y no sólo se pierda lo aprendido, sino que además se entorpezca el aprendizaje del otro módulo.

## 5.4. Style Transfer

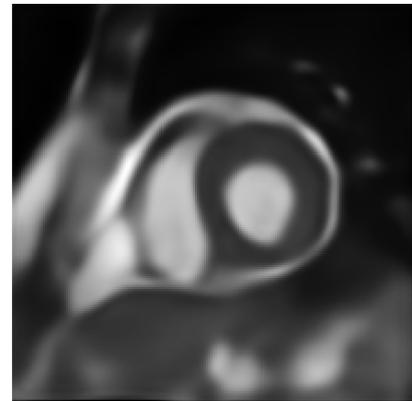
La transferencia de estilo ha resultado ser un método relativamente sencillo y eficaz, con el que se ha podido mejorar sustancialmente el resultado obtenido hasta ahora con el CVAe. Gracias a esta técnica, hemos conseguido solventar la diferencia principal que tenían las imágenes que estábamos generando con las reales, dotándolas de las texturas necesarias para tener ese aspecto realista.

Tenemos en primer lugar los resultados obtenidos utilizando la red VGG19, seguidos por los resultados utilizando el *encoder* del CVAe. Finalmente tenemos los resultados de la herramienta de visualización.

Los resultados que se muestran provienen de aplicar el estilo de la figura 21a a la figura 21b.



(a) Imagen real de la que se obtiene el estilo.



(b) Imagen generada a la que se le aplica el estilo

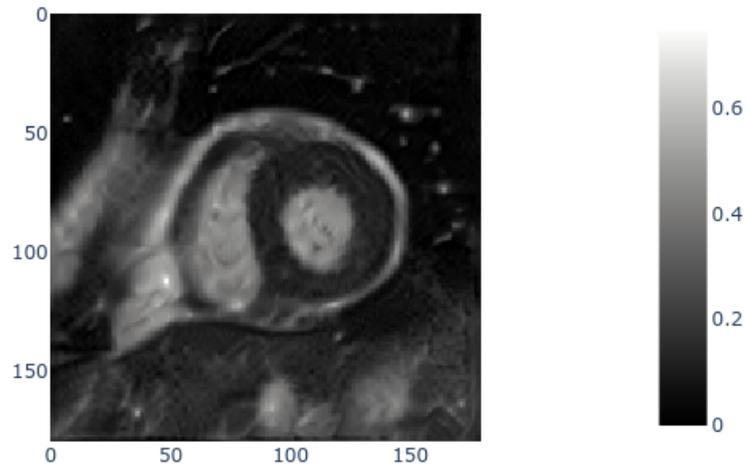
Figura 21: Imágenes utilizadas para los resultados mostrados en la transferencia de estilo.

### 5.4.1. Resultados con VGG19

Aunque los resultados aún distan de tener un aspecto completamente realista, mediante la transferencia de estilo conseguimos una mejora importante. Con esta técnica podemos obtener un resultado con texturas muy similares a las originales a costa de tener ligeros artefactos, como observamos en la figura 22a, o resultados sin artefactos a costa de no tener unas texturas tan realistas, como se puede apreciar en la figura 22b.

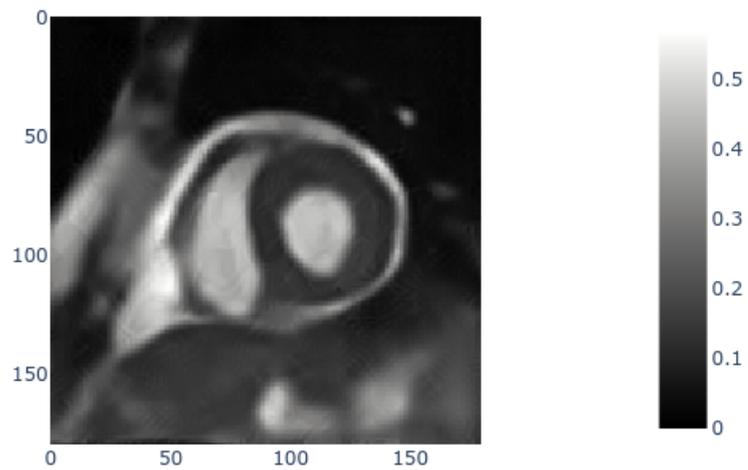
Si bien es cierto que para la correcta aplicación de esta técnica resulta beneficioso tener una imagen original similar a la cual le queremos aplicar el estilo, esto no resulta un problema ya que podemos encontrar una imagen original similar a la que buscamos haciendo uso del espacio latente del CVAe. Dado que tenemos un codificador para mapear imágenes a dicho espacio latente, podemos tener fácilmente la posición que una

LR: 5e-02 | SW: 1e-09 | CW: 1e-10



(a) Learning rate 5e-02; Style weight 1e-09; Content weight 1e-10

LR: 5e-05 | SW: 1e-06 | CW: 1e+03



(b) Learning rate 5e-05; Style weight 1e-06; Content weight 1e+03

Figura 22: Resultados de aplicar la transferencia de estilo con el modelo VGG19.

imagen original ocupa en dicho espacio. De igual forma, para generar una nueva imagen, partimos de un vector aleatorio que indica la posición de la imagen a generar en el mismo espacio latente. Es por ello que podemos fácilmente encontrar la imagen original adecuada para aplicar la técnica calculando la distancia entre la imagen generada y las originales, y seleccionando la imagen original más cercana en dicho espacio latente.

#### 5.4.2. Resultados con el *encoder* del CVAe

La utilización del propio *encoder* del CVAe como modelo mediante el cual realizar la transferencia de estilo no ha resultado satisfactoria. Esto puede deberse en parte a que el *encoder* no ha sido capaz de capturar correctamente las características de alta frecuencia de las imágenes, lo que implica que dichas características de alta frecuencia no están correctamente representadas en el espacio latente, y por tanto el *decoder* no puede contar con la información suficiente para poder generar las texturas necesarias. Podemos ver un ejemplo de la transferencia de estilo con las capas del *encoder* en la figura 23.

LR: 5e-02 | SW: 1e-02 | CW: 1e-10

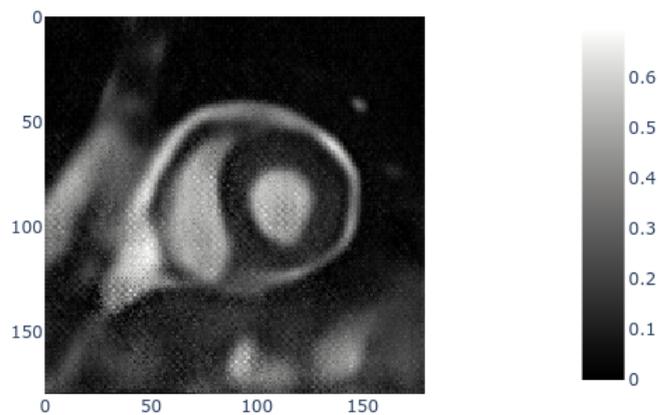


Figura 23: Resultados de aplicar la transferencia de estilo con el *encoder* de nuestro CVAe.

#### 5.4.3. Dashboard

En cuanto a la herramienta de visualización creada, no sólo ha resultado ser de extrema ayuda para poder ver de forma rápida y cómoda el conjunto de imágenes, sino que además ha permitido poder comprobar rápidamente y de forma intuitiva el funcionamiento de la transferencia de estilo, permitiendo entender de forma sencilla el efecto que tienen los diferentes parámetros al poder interactuar con ellos y visualizar en el momento lo que supone su variación. Podemos ver el resultado del dashboard en la figura 24, así como en el enlace a la aplicación (<https://fierce-bayou-68989.herokuapp.com/>).

### TransferStyle Results

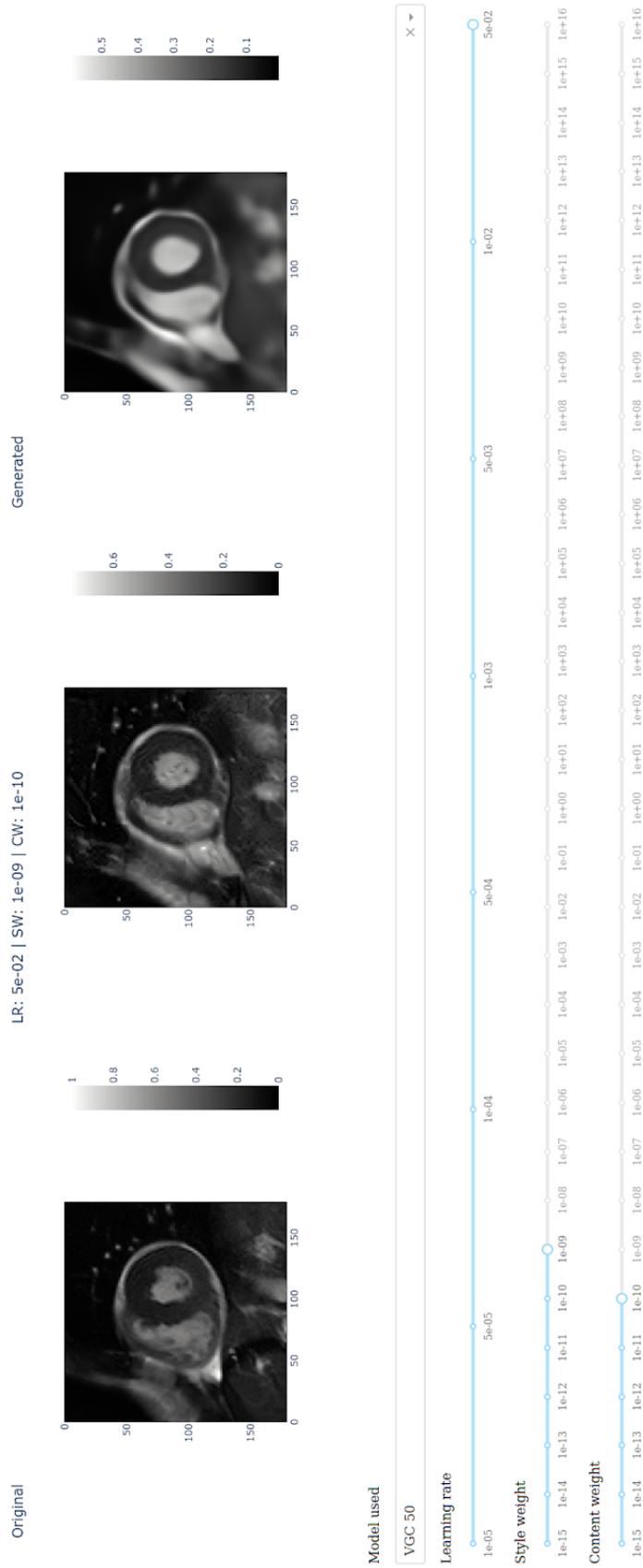


Figura 24: Herramienta de visualización de la transferencia de estilo.

## 6. Conclusiones

El objetivo de este trabajo era generar imágenes de resonancia cardíaca que pudieran servir para aumentar conjuntos de datos reales. Se han probado diferentes tipos de arquitecturas basadas en tres tipos de metodologías de modelos generativos (VAE, CVAe, GAN), obteniendo un generador (CVAe) capaz de crear imágenes cuya estructura anatómica general es muy realista, pero que les faltan los detalles propios de las texturas de las imágenes de resonancia cardíaca. Mediante la transferencia de estilo se consigue dotar en gran parte de ese nivel de detalle por lo que juntando el generador CVAe junto a la metodología desarrollada de transferencia de estilo, hemos logrado tener una herramienta de generación de imágenes de resonancia magnética.

Además, durante la realización del trabajo, se ha desarrollado una herramienta de visualización que no solo nos ha permitido poder ver de forma sencilla las imágenes, sino que ha resultado ser una potente herramienta a la hora de entender el funcionamiento de la transferencia de estilo, y sobre todo la importancia que tienen sus parámetros.

### 6.1. Líneas futuras

- Explorar con mayor detalle la GAN y las muchas variantes que ofrece.
- Mejorar el CVAe, de forma que se fuerce a que utilice parte de las dimensiones del espacio latente a contener información sobre las altas frecuencias.
- Utilizar técnicas de extracción de características de imágenes tal como la transformada de wavelet para forzar al modelo a que aprenda las altas frecuencias de la imagen.
- Explorar técnicas alternativas de generación tal como Normalizing Flow.
- Ampliar las funcionalidades de la herramienta de visualización de las imágenes de la transferencia de estilo.

# 7. Publicaciones derivadas de este TFM

J.M. Jaén-Lorites, M. Pérez-Pelegri, V. Laparra, D. Moratal. *Understanding the style transfer deep-learning technique through a web dashboard.*

Proceedings of the 16th Annual International Technology, Education and Development Conference, INTED, 2022. (Abstract aceptado, trabajo enviado)

**UNDERSTANDING THE STYLE TRANSFER DEEP-LEARNING TECHNIQUE THROUGH A WEB DASHBOARD**

J.M. Jaén-Lorites<sup>1</sup>, M. Pérez-Pelegri<sup>2</sup>, V. Laparra<sup>2</sup>, D. Moratal<sup>1</sup>  
<sup>1</sup>Universitat Politècnica de València (SPAIN)  
<sup>2</sup>Universitat de València (SPAIN)

**Abstract**

In deep learning, one of the main fields in artificial intelligence, every day arises a new technique. Style transfer is one of these newer techniques with which you can take the style of an image, for example the style of The Starry Night of V.M.J. van Gogh, and apply it to another image, for example the Mona Lisa, resulting in a new image with the Mona Lisa as if Van Gogh had painted it.

What the researchers in these techniques found is that not only they need time to understand them but they also have numerous parameters and options that can be tuned, affecting to the final result.

The aim of this work is to develop an interactive tool where you can see the effect of different parameters related to the artificial intelligence (deep learning) Style Transfer technique in a simple and intuitive way. The idea is to allow users to play with the different parameters of the Style Transfer technique and see how they influence the result of its application. This allows users to visualize in a simple way how these parameters relate to each other, and how they affect the transfer of the style. Specifically, the technique is applied to cine cardiac magnetic resonance images which are a type of medical image.

**Keywords:** educational tool, deep learning, image analysis, Style Transfer, dashboard, image processing, medical imaging, artificial intelligence

**1 INTRODUCTION**

Artificial intelligence (AI) is one of the fastest-growing fields in recent years [1]. AI is the science that seeks to mimic human behavior in some way. Within AI there is the field of machine learning (ML). ML involves a set of techniques that allow machines to learn from data, obtaining AI applications. The obtained results depend heavily on the data, and it is necessary in many cases to perform preprocessing to obtain the intrinsic features of the data. This is where deep learning (DL) arises, where the machine not only learns to use the data but also to extract its features from the data beforehand. It is called "deep" because it uses a technique called neural networks, with many layers, that is, a lot of depth [2].

One of the best-known examples of DL is image classification. A classic use case is to detect whether there is a cat or a dog in a picture. Another example is the generation of synthetic data or images, such as deep fakes, where you generate images that look real but are not. There is also Style Transfer in the category of image generation [3].

Style Transfer is a deep learning technique where the style of an image is applied to another image from which the content is taken. That is, certain features of an image, such as textures or color, are transferred to another image from which particular information such as subjects or the scene is retained. This allows, for example, to obtain an image of the Mona Lisa as if it had been painted by Van Gogh with The Starry Night style as shown in Fig. 1. Three factors have a great influence on this and are closely related to each other: These are learning rate, style weight, and content weight.

The style weight, in the case of the Mona Lisa painted by Van Gogh, would indicate the freedom that Van Gogh has to apply his famous style. On the other hand, the content weight would be related to how much it should still resemble the original Mona Lisa in terms of content, that is, how important it is to keep all the objects and general structure of the painting. Finally, the learning rate has an effect on the learning speed, which in practice means a different result in terms of the smoothness of the generated image [4].

J.M. Jaén-Lorites, M. Pérez-Pelegri, V. Laparra, M.P. López-Lereu, J.V. Monmeneu, A.M. Maceira, D. Moratal. *Synthetic Generation of Cardiac Magnetic Resonance Images Combining Convolutional Variational Autoencoders and Style Transfer.*

Proceedings of the 44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2022. (Trabajo enviado)

**Synthetic Generation of Cardiac Magnetic Resonance Images Combining Convolutional Variational Autoencoders and Style Transfer**

José Manuel Jaén-Lorites, Manuel Pérez-Pelegri, Valero Laparra, María Pilar López-Lereu, José Vicente Monmeneu, Alicia M. Maceira, and David Moratal, *Senior Member, IEEE*

**Abstract**—The number of studies in the medical field that use machine learning and deep learning techniques has been increasing in the last years. However, these techniques require a huge amount of data that can be difficult and expensive to obtain. This specially happens with cardiac magnetic resonance images. One solution to the problem is to use the dataset used to generate synthetic data. Convolutional Variational Autoencoder (CVAE) is a deep learning technique which allows to generate synthetic images but sometimes the synthetic images can be slightly blurred. We propose the combination of the CVAE technique and Style Transfer technique to generate synthetic realistic images.

**Clinical Relevance**—The current work presents a tool to generate synthetic data in a simple, easy, and fast way the cardiac magnetic resonance images dataset with which perform machine learning and deep learning studies.

**1. INTRODUCTION**

Artificial intelligence (AI) is one of the fastest-growing fields in recent years [1]. AI is the science that seeks to mimic human behavior in some way. Within AI there is the field of machine learning (ML). ML involves a set of techniques that allow machines to learn from data. Deep Learning (DL) is a ML subfield that uses Artificial Neural Networks to solve problems such as recognition or classification, and has become very popular because of its great success [2]. This also happens in the medical field, where the number of studies related to the use of machine learning has doubled almost every year since 2015. Two-hundred of the 2000-2020 publications related to deep learning in medical field has been released between 2019 and 2020 [3]. The use and success of the deep learning in the medical field is unquestionable.

To use deep learning techniques, usually a huge amount of data is needed. However, medical data is related to many protection data laws in a lot of countries and needs to be anonymized. This process of anonymization involves resources as money and time. That makes the medical data available to perform studies very low and affects directly the capacity of keep having successful ML studies related to medical fields.

To solve this, the use of deep learning data generation techniques is proposed in order to have a tool with which generate synthetic data in a simple, easy and fast way without spend resources obtaining or anonymizing it. There are studies where already have generated medical images such as X-ray chest images or histology images [4].

The aim of this work is to generate cardiac magnetic resonance images in order to enlarge the available dataset in a simple, easy and fast way using deep learning data generation technique (Convolutional Variational Autoencoder (CVAE). Additionally, we propose the use of Style Transfer technique to improve the quality of the generated data.

**II. BACKGROUND**

**A. Autoencoder**

An autoencoder is a type of neural network architecture in which two separate modules are differentiated. These modules are the encoder and the decoder. They generally have a mirror architecture, where the encoder is responsible for encoding the incoming information in a latent space of smaller dimension than the input data, and the decoder must, from the information encoded in this latent space, be able to recover the original data.

**B. Variational Autoencoder**

The Variational Autoencoder (VAE) is a special type of autoencoder, which can be interpreted as a probabilistic autoencoder. That is, its outputs are partially determined by a stochastic component, even in the inference. They owe their name to the fact that they make use of variational Bayesian inference.

The main difference with normal autoencoders is that the encoder does not directly produce an encoding of the data in the latent space, but produces a mean  $\mu$  and a standard deviation  $\sigma$ , so that the final encoding is randomly drawn from a normal distribution with mean  $\mu$  and standard deviation  $\sigma$ .

In this way, the output does not belong to the encoding of the input, but to the encoding of data that are very close in latent space to the input data, and therefore are expected to be very similar to the input data [5].

30

## Referencias

- [1] J G Planas. Introducción Biofísica a la Resonancia Magnética en Neuroimagen. *V:03-2*, pages V:03–2, 2003.
- [2] Arthur Guyton and John Hall. *Guyton y Hall Tratado de Fisiología Médica*. 2011.
- [3] C Hernández, B Zudaire, S Castaño, P Azcárate, A Villanueva, and G Bastarrika. Principios básicos de resonancia magnética cardiovascular (RMC): secuencias, planos de adquisición y protocolo de estudio. *Anales del Sistema Sanitario de Navarra*, 30(3), dec 2007.
- [4] A.Géron. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 2019.
- [5] Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.
- [6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [7] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, jan 2015.
- [8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.
- [9] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.
- [10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27*, jun 2014.
- [11] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, jun 2016.
- [12] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. GANs for Medical Image Synthesis: An Empirical Study. pages 1–24, 2021.