

**MÁSTER UNIVERSITARIO EN
INGENIERÍA DE TELECOMUNICACIÓN**

TRABAJO FIN DE MÁSTER

***ANÁLISIS Y PROPUESTA DE SOLUCIONES
BASADAS EN IA PARA LA CANCELACIÓN DE
AUTOINTERFERENCIAS EN TRANSMISORES IN-
BAND FULL DUPLEX***

Estudiante	<i>Llaguno, Perez, Endika</i>
Director	<i>Angueira, Buceta, Pablo</i>
Codirector	<i>Montalban, Sanchez, Jon</i>
Departamento	<i>Departamento de comunicaciones</i>
Curso académico	<i>2021/2022</i>

Bilbao, 17 de septiembre 2022

Resumen

El desarrollo de nuevos estándares para la próxima generación de radiodifusión, concretamente el estándar ATSC 3.0, supondrá una gran innovación en los sistemas de radiodifusión. Este estándar presenta un nuevo tipo de tecnologías, conocidas como *In-Band Distribution Link* (IDL) e *Inter-Tower Network* (ITC), estas pretenden cambiar la arquitectura tradicional de la red ATSC introduciendo numerosas ventajas. A pesar de ello, estas tecnologías presentan una gran desventaja, la generación de una señal autointerferente en las torres de difusión de la red. Esta señal es generada debido al uso del mismo canal de frecuencia en recepción y transmisión. Existen técnicas tradicionales de procesamiento de señal para la cancelación de señales de autointerferencia, pero estas no son óptimas y tampoco se adecuan a las características concretas de este sistema de comunicaciones. Por ello se ha presentado en este proyecto un estudio de nuevas soluciones para la cancelación de señales autointerferentes basadas en Inteligencia Artificial. Concretamente, se ha utilizado un nuevo tipo de red neuronal conocida como *Vision Transformer* para la resolución del problema. Estas redes diseñadas para tareas de visión artificial han sido adaptadas para la resolución del problema de cancelación de autointerferencia presentado.

Palabras clave: Autointerferencia, In-Band Full Duplex, Inteligencia Artificial, Vision Transformer.

Laburpena

Irrati-difusioaren hurrengo sorkuntzarako estandarrak garatzeak, zehazki ATSC 3.0 estandarra, berrikuntza handia ekarriko luke irrati-difusio sistemetan. Estandar honek teknologia mota berri batzuk aurkezten ditu, In-Band Distribution Link (IDL) eta Inter-Tower Network (ITC) bezala ezagutzen direnak. Hauek, ATSC sarearen arkitektura tradizionala aldatu nahi dute, abantaila ugari sartuz. Hala eta guztiz ere, teknologia horiek desabantaila handi bat dute, sareko difusio-dorreetan seinale autointerferente bat sortzen baitute. Seinale hori harreran eta transmisioan maiztasun-kanal bera erabiltzeagatik sortzen da. Autointerferentzia-seinaleak bertan behera uzteko hainbat seinale prozesamendu teknika tradizionalak daude, baina teknika hauek ez dira optimoak, eta ez dira komunikazio-sistema honen ezaugarri zehatzetara egokitzen. Hori dela eta, proiektu honetan Adimen Artifizialean oinarritutako deuseztapenerako irtenbide berrien azterlan bat aurkeztu da. Zehazki, arazoa konpontzeko Vision Transformer izeneko sare neuronal mota berri bat erabili da. Ikusmen artifizialeko zereginetarako diseinatutako sare hauek aurkeztutako autointerferentzia bertan behera uzteko arazoa konpontzeko egokitu egin dira.

Gako-hitzak: Autointerferentzia, In-Band Full Duplex, Adimen Artifiziala, Vision Transformer.

Summary

The development of new standards for the next generation of broadcasting, specifically the ATSC 3.0 standard, will be a major innovation in broadcasting systems. This standard introduces a new type of technologies, known as In-Band Distribution Link (IDL) and Inter-Tower Network (ITC), these are intended to change the traditional architecture of the ATSC network introducing numerous advantages. However, these technologies have a major disadvantage, the generation of a self-interfering signal in the network's broadcast towers. This signal is generated due to the use of the same frequency channel in reception and transmission. There are traditional signal processing techniques for the cancellation of self-interference signals, but these are not optimal and are not suitable for the specific characteristics of this communication system. Therefore, a study of new solutions for cancellation based on Artificial Intelligence has been presented in this project. Specifically, a new type of neural network known as Vision Transformer has been used to solve the problem. These networks designed for artificial vision tasks have been adapted to solve the self-interference cancellation problem presented.

Keywords: Self-interference, In-Band Full Duplex, Artificial Intelligence, Vision Transformer.

Índice

1. Antecedentes y motivación	8
2. Objetivos	13
3. Metodología	14
3.1. Resumen de la metodología	14
3.2. Definición del problema	15
3.3. Estado del arte	15
3.3.1. Estado del arte Procesado de Señal	16
3.3.2. Estado del arte Inteligencia Artificial	16
3.4. Estudio de las transformers	16
3.5. Diseño e implementación del sistema de cancelación	16
3.6. Pruebas y validación del sistema	17
3.7. Análisis de rendimiento	17
4. Definición del problema	18
5. Análisis del estado del arte	20
5.1. Procesado de señal	20
5.1.1. Ecualización	20
5.1.2. Gestión dinámica de ganancia	21
5.1.3. Cancelación en dos pasos mediante filtrado digital	22
5.2. Inteligencia Artificial	23
5.2.1. Redes convolucionales	23
5.2.2. LSTM	31
5.2.3. Transformers	33
6. Estudio de alternativas de IA para cancelación	37

7. Transformers como solución al problema	38
7.1. Restormer	38
8. Pruebas y validación	40
8.1. Despliegue y configuración de la plataforma SW	40
8.2. Configuración de la red <i>Restormer</i>	40
8.3. Generación y estructuración de los dataset	41
8.4. Aproximación 1: PNG blanco/negro	43
8.5. Aproximación 2: PNG a color	44
8.6. Aproximación 3: Matrices como dato de entrada	47
8.6.1. Escenario 1: Canales sin doppler	47
8.6.2. Escenario 2: Canal directo con doppler	50
8.6.3. Escenario 3: Canal directo y de Loopback con doppler	51
8.6.4. Mejoras a futuro	52
9. Plan de proyecto	54
9.1. Paquetes de trabajo y tareas	54
9.2. Diagrama de Gantt	56
10. Conclusiones	58
11. Bibliografía	59

Índice de figuras

1.	Arquitectura tradicional de la red ATSC	9
2.	Arquitectura de la red ATSC 3.0 con IDL	9
3.	Esquema de las torres de una red basada en tecnología IDL/ITC	10
4.	Bloques empleados en la cadena de recepción	11
5.	Resumen de la metodología utilizada en el proyecto.	15
6.	Técnica de ecualización para la estimación de canal	20
7.	Bucle de ajuste dinámico de ganancia	21
8.	Esquema de cancelación en dos fases	22
9.	Arquitectura de una red convolucional	23
10.	Operación de convolución para la obtención de un mapa de líneas verticales	24
11.	Arquitectura de una red CNN autoencoder	25
12.	Arquitectura de la red SRCNN	26
13.	Arquitectura de la red VGG16 net	28
14.	Comparación entre los bloques de una transformer, una CNN (ResNet) y una red ConvNext	30
15.	Arquitectura de un bloque LSTM	31
16.	Arquitectura de una red RNN	33
17.	Mapa de atención de un problema de NLP	34
18.	Bloques de una red transformer	35
19.	Arquitectura de una ViT	36
20.	Bloques de la red <i>Restormer</i> [31]	39
21.	Generación de la matriz de entrada a la red	42
22.	Datos de entrada a la red, (a) dato de entrada y (b) etiqueta del dato . . .	44
23.	Señal estimada y canal de loopback para datos de entrada PNG <i>grayscale</i> .	44
24.	Proceso de creación de las imágenes a color	45
25.	Comparación entre la estimación de la red y canal de loopback	45

26.	MSE de la señal estimada comparada con la señal original	46
27.	Señal estimada, canal de loopback y señal de entrada a la red	47
28.	Comparación entre las partes reales e imaginarias de las señales	48
29.	Señal estimada, canal de loopback y señal de entrada a la red tras la corrección	48
30.	Análisis del MSE del escenario 1	49
31.	Señal estimada y canal de loopback (escenario 2)	50
32.	Análisis del MSE del escenario 2	51
33.	Señal estimada y canal de loopback (escenario 3)	51
34.	Análisis del MSE del escenario 3	52
35.	Zoom en la señal de MSE	53
36.	Diagrama de Gantt	57

Índice de tablas

1.	Ventajas y desventajas de las técnicas de IA	37
2.	Parámetros de entrenamiento de la red	41

1. Antecedentes y motivación

En los últimos años, la industria de la radiodifusión se ha estado enfrentando a numerosos retos debido al rápido incremento de la demanda de contenidos comerciales a través de sistemas broadband wireless [1]. Por otro lado, las bandas por debajo de 1 GHz de frecuencia, como por ejemplo la banda UHF (Ultra High Frequency) donde se encuentra la transmisión de la televisión digital terrestre, ofrecen numerosas ventajas frente a otras bandas de frecuencia superior, como por ejemplo: menos pérdidas debidas a la propagación, alta capacidad de penetración, menor coste de los equipos...

Es por ello que el Advanced Television Systems Committee (ATSC) comenzó a desarrollar un nuevo estándar para la próxima generación de sistemas digitales de radiodifusión, es conocido como ATSC 3.0. Con ATSC 3.0 se conseguirá una mayor capacidad de datos, un funcionamiento mucho más robusto, un uso del espectro más eficiente y el despliegue de nuevos tipos de servicio en este tipo de sistemas. Para ello, ATSC 3.0 propone una arquitectura de red de radiodifusión diferente a la empleada tradicionalmente. Esta nueva arquitectura dispondrá de una red troncal, conocida como *Core*, y de una red de acceso vía radio (RAN). Esta RAN será una red unidireccional con el usuario final, si bien incluye enlaces de comunicaciones con otros nodos de dicha red (otros transmisores). Uno de los requerimientos de esta arquitectura es la existencia de una red de comunicación, conocida como *backhaul*, la cual es la encargada de la interconexión de la red de difusión. Una de las propuestas que se pretende introducir en el estándar ATSC 3.0 es el uso de técnicas de multiplexación LDM para realizar la red de *backhaul* en la misma banda en la que se transmite la señal de difusión [1].

De este modo, las funciones de red de transporte, de comunicación entre centros y de difusión de la señal a los usuarios finales se hará dentro del mismo canal de la banda de frecuencia UHF.

Esta propuesta da lugar a un nuevo tipo de tecnologías conocidas como IDL (In-Band Distribution Link) [2] e ITC (Inter Tower Communication) [3]. Una de las principales ventajas de este esquema de red es que ya no es necesario un link entre el proveedor del contenido *broadcast* y cada una de las torres de transmisión, conocido con enlace STL (Study to Transmitter Link). Este link es uno de los elementos que mayor coste tiene en las redes de radiodifusión, debido a que suele ser un enlace dedicado llevado normalmente mediante fibra óptica o un radioenlace dedicado. En este nuevo tipo de redes es suficiente con llevar un único enlace STL a unas de las torres y que sea esta la que mediante la red de *backhaul* se encargue de distribuirlo a sus torres vecinas. Del mismo modo, estas torres se lo distribuirán a sus vecinas, reduciendo así el número de enlaces STL considerablemente.

Las tecnologías IDL/ITC se basan en el uso de técnicas de multiplexación no ortogonales, concretamente en la técnica LDM (Layered Division Multiplexing) [4]. Mientras que las técnicas de multiplexación tradicionales TDM (Time Division Multiplexing) y FDM (Frequency Division Multiplexing), se basan en la división de los recursos de tiempo-frecuencia del canal, las técnicas LDM hacen un uso completo de todos estos recursos. En los sistemas LDM se utiliza una estructura de transmisión por capas donde se transmiten simultáneamente múltiples señales con diferentes niveles de potencia, correspondientes a

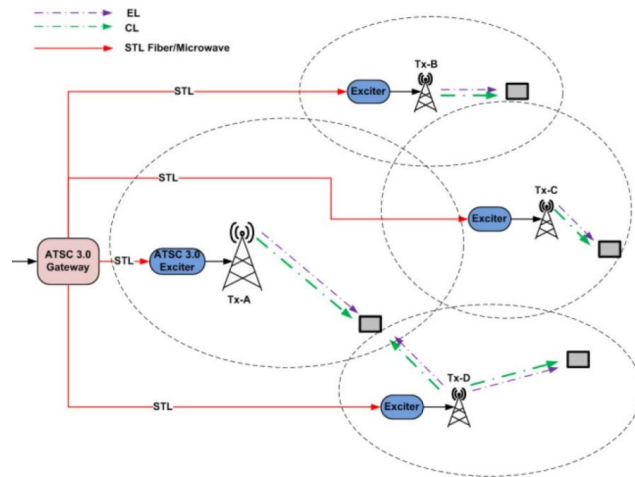


Figura 1: Arquitectura tradicional de la red ATSC.

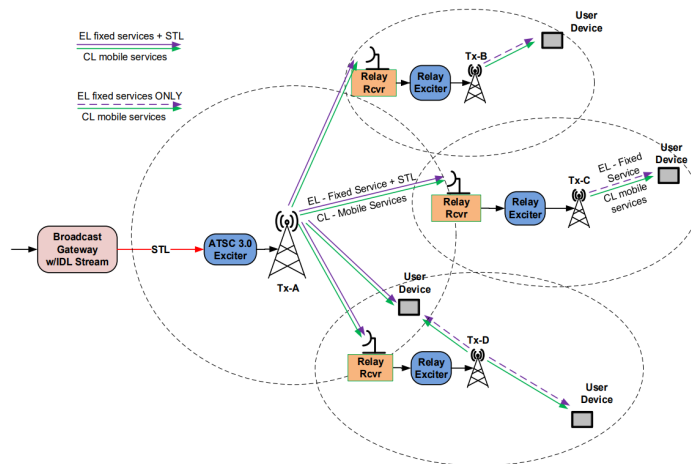


Figura 2: Arquitectura de la red ATSC 3.0 con IDL

diferentes tipos de servicios [4]. En el caso de uso presentado en este proyecto se emplean dos capas de servicios diferenciados, una de ellas siendo la capa de mayor potencia, donde se transmite la señal de difusión, y la de menor potencia, donde se envía la señal de *backhaul* mencionada anteriormente.

Las redes basadas en tecnología IDL/ITC tienen un gran número de ventajas, como ya se ha explicado, pero también cuentan con varios inconvenientes que deben ser solucionados para una buena implementación de este tipo de redes. La principal desventaja es la señal de autointerferencia que se genera en las torres de reemisión, la cual es producto del acople de la señal de transmisión en recepción, y viceversa, debido al uso del mismo canal de frecuencia en ambos puntos.

En la *Figura 3* se observa el esquema de estas torres y como se genera la señal de autointerferencia, a partir de ahora llamada *loopback*. Por tanto, la señal recibida ya no será la señal deseada, sino que está compuesta por la señal deseada (conocida como *forward*), la señal interferente *loopback* y el ruido que se genera en un sistema de comunicaciones

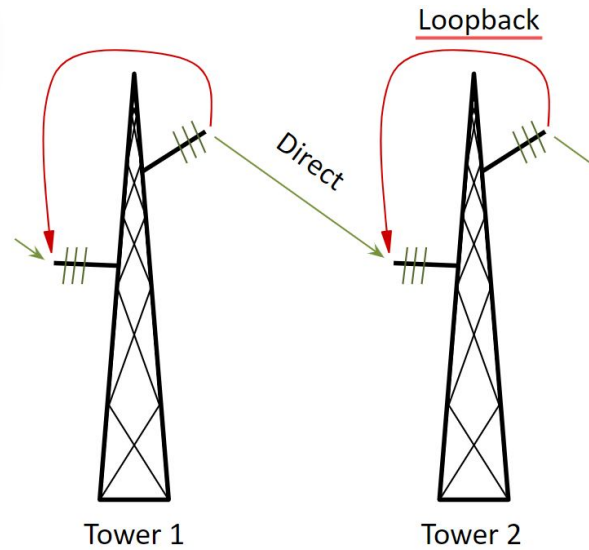


Figura 3: Esquema de las torres de una red basada en tecnología IDL/ITC

inalámbrico. La composición de esta señal recibida puede observarse en la *Ecuación 1*, donde x_{FW} es la señal deseada, h_{FW} es el canal entre la torre y su vecina, x_{LB} es la señal de loopback, h_{LB} el canal de loopback y n el ruido intrínseco a un sistema de telecomunicaciones.

$$y_{Rx} = x_{fw} * h_{fw} + x_{lb} * h_{lb} + n \quad (1)$$

La señal que se desea obtener es x_{FW} , es por ello, por lo que se deberá de eliminar el resto de los elementos que componen la señal recibida. El primer paso es el aislamiento entre las antenas receptora y transmisora. Debido a que se trata de un sistema broadcast donde la señal de difusión debe enviarse con una PRA (Potencia Radiada Aparante) alta, el aislamiento entre Rx y Tx será limitado, por lo que será necesario añadir algún tipo de cancelación adicional a este sistema. Tradicionalmente, se ha empleado una cadena de recepción para la obtención de la señal compuesta por un bloque de *beamforming*, una cancelación analógica y una cancelación digital [5]. Debido a que la señal de loopback llega con mayor potencia que la señal deseada, es necesario tener un nivel muy alto de cancelación. En el esquema tradicional, el primer paso que se realiza es el aislamiento entre antenas, con el que se consigue una cancelación de unos 30 dBs. Tras ello se emplea la cancelación analógica, con lo que se pueden obtener hasta unos 50 dBs de cancelación. Por último, se emplean técnicas de cancelación digital donde se consiguen unos 20 dBs de cancelación [6]. Con estas técnicas se consigue una estimación bastante precisa de la señal x_{FW} .

Estas técnicas planteadas hasta ahora presentan algoritmos de cancelación estáticos o aproximaciones simplificadas debido a que:

- No consideran las no linealidades entre los elementos de transmisor y receptor [7].
- No consideran la distorsión no lineal asociada a la propagación.
- No tienen en cuenta diferentes fuentes de ruido como el ruido de cuantificación ni el ruido de fase [8]

- Por último, las técnicas tradicionales tienden a no tener en cuenta la variabilidad temporal del canal de loopback. En los sistemas reales, la variabilidad temporal del canal de loopback ha de tenerse en cuenta y la precisión de esta estimación va a determinar en gran medida la efectividad de la cancelación.

Además, muchos de los algoritmos de cancelación digital, como los algoritmos LS (Least-Squares) o el MMSE (Minimum Mean Square Error) son modelos matemáticos demasiado complejos como para ser empleados en soluciones en tiempo real.

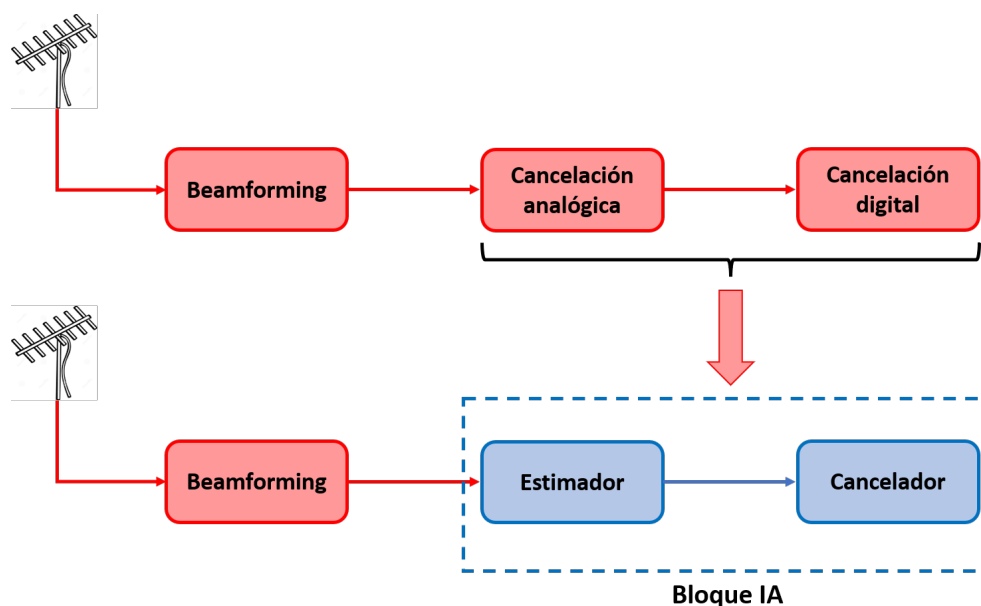


Figura 4: Bloques empleados en la cadena de recepción

Por todas estas razones se decidió estudiar técnicas alternativas para la cancelación de la señal autointerferente. La nueva línea de investigación seguida en el proyecto se ha basado en el estudio de técnicas basadas en IA. Con ellas se pretende conseguir una buena estimación del canal de loopback, la cual se restará a la señal recibida, junto con la señal x_{LB} . De este modo se anula la componente interferente en recepción, dando como resultado un esquema tradicional de comunicaciones donde se tiene la señal de interés convolucionada con un canal y un ruido. Debido a esto, el esquema de recepción cambiará, siendo el nuevo esquema de recepción el que se observa en *Figura 4*. Debido a que el canal de loopback es desconocido, el uso de IA puede no parecer la solución más óptima, pero este canal a pesar de ser desconocido cumple ciertas características de las señales naturales, lo cual hace que el uso de IA sea muy adecuado [9]. Estas características son:

- **Localidad:** Las muestras cercanas de la señal guardan mayor correlación que las lejanas.
- **Composicionalidad:** La forma de la señal se encuentra compuesta por patrones más pequeños, los cuales son característicos e identificables.
- **Estacionaridad:** La señal se repite cada cierto tiempo, es decir, se pueden tomar dos imágenes del canal y se observa que se repite periódicamente.

Hace 10 años, con el auge de la IA, hubo un intento de solucionar los problemas de estimación y cancelación de canal con esta tecnología [10] [11] [12] [13]. A pesar de ello, no existe una solución óptima para resolver el problema de autointerferencia. Por ello, en este proyecto se ha decidido estudiar diferentes técnicas y encontrar una que sea óptima para la resolución del problema.

2. Objetivos

El objetivo de este TFM es el estudio de técnicas de IA para la resolución del problema de la autointerferencia en reemisores ATSC 3.0 con tecnología In-Band Distribution Link (IDL). En estas redes el equipo reemisor debe operar en la misma frecuencia en transmisión y recepción, por lo tanto, es necesario cancelar el acoplamiento que se produce entre la antena receptora y la antena transmisora. Para este tipo de cancelación existen soluciones basadas en técnicas tradicionales como el procesado de señal. Es por ello que en este proyecto se han buscado nuevas soluciones, concretamente, basadas en IA.

Para conseguir este objetivo, se han fijado varios objetivos intermedios que deberán cumplirse:

- Disponer de un modelo del problema de autointerferencia. Obtención de la definición analítica del problema, descripción matemática, bases de datos...
- Disponer de un análisis de las diferentes técnicas de IA que se pueden emplear para la resolución del problema.
- Disponer de un análisis de las diferentes arquitecturas de redes neuronales que se pueden emplear para la resolución del problema.
- Selección de una de las arquitecturas.
- Disponer de un entorno de prueba y validación de la solución propuesta. Este entorno deberá permitir validar y ajustar el rendimiento de la red. Para ello será necesario la definición de métricas que permitan la comparación de la solución propuesta, con las técnicas tradicionales de procesado de señal.

3. Metodología

3.1. Resumen de la metodología

El primer paso para realizar este proyecto ha sido la definición concreta del problema que se desea resolver. Se comienza definiendo analíticamente las matemáticas del problema y estableciendo las ecuaciones sobre las que se necesitará trabajar, así como los términos de interés dentro de la ecuación. También será necesario la obtención de los conjuntos de datos que se van a emplear para la simulación de las soluciones.

Tras dicha definición se ha realizado un estudio del estado del arte de las técnicas de procesamiento de señal empleadas tradicionalmente como solución al problema de cancelación. Una vez se han estudiado dichas técnicas se ha pasado al estudio de soluciones más novedosas, concretamente, técnicas basadas en IA. Tras estos estudios se ha seleccionado la familia de IA a emplear en la resolución del problema.

Esta selección nos ha llevado a realizar un estudio de las diferentes arquitecturas de redes existentes dentro de esta familia. Tras ello se ha decidido cuál será la arquitectura a emplear y se ha comenzado con el despliegue del software necesario para su correcto funcionamiento. Se ha implementado esta solución, utilizando el diseño adecuado para el problema que debemos de resolver. Se ha validado el correcto funcionamiento de la plataforma software empleada y se ha pasado a realizar la evaluación y pruebas de la solución seleccionada.

Por último, se ha realizado un análisis de rendimiento de la solución y se han extraído las conclusiones sobre el proyecto.

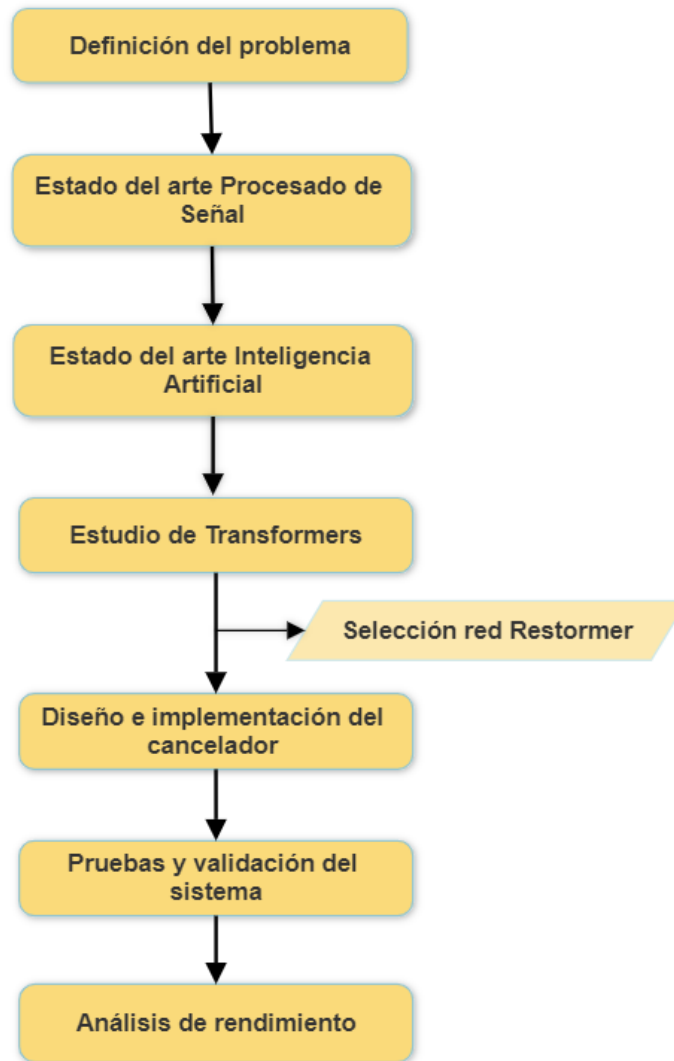


Figura 5: Resumen de la metodología utilizada en el proyecto.

3.2. Definición del problema

Definición de la estructura de la arquitectura de las torres de reemisión, así como de las características de las comunicaciones que se van a emplear. Se realiza el modelado matemático de las señales recibidas en estas torres, así como los términos de interés de dichas ecuaciones. Se estudiarán maneras de poder simplificar las ecuaciones para facilitar la solución del problema. También se van a establecer las características del problema, empleando estas realizar una lista de alternativas que puedan servir como solución.

3.3. Estado del arte

Estudio detallado de los precedentes del proyecto, se realiza para entender el contexto y establecer el conocimiento propio sobre el tema. Sirve para realizar un estudio del ámbito

en el que se establecerá el proyecto, estudiar las soluciones ya existentes y cómo pueden mejorarse, los beneficios que se podrían aportar con una nueva solución y descubrir las herramientas adecuadas para la resolución del problema, así como los recursos disponibles.

3.3.1. Estado del arte Procesado de Señal

El estudio en profundidad de las técnicas ya existentes para la resolución de la cancelación de autointerferencia. Repaso de los métodos utilizados en equipos que sufran de este problema, analizando las ventajas y desventajas de cada uno de los métodos, así como de la viabilidad de utilizarlos en los equipos reemisores de este estudio. Con este estudio también se pretende ampliar los conocimientos que se tienen del problema de autointerferencia, además de aprender nuevos modelos matemáticos para su resolución. Por último, permitirá obtener un método estandarizado con el que se compararán los resultados obtenidos por la solución propuesta en este proyecto.

3.3.2. Estado del arte Inteligencia Artificial

Estudio en detalle las diferentes técnicas de IA existentes hoy en día. En este estudio se pretende seleccionar una de estas soluciones. Para ello, se han investigado las técnicas ya existentes filtrándolas con los criterios seleccionados que deberán cumplir para poder resolver el problema. También se han analizado las arquitecturas de estas redes, así como si deberían de ser modificadas para poder ser empleadas.

3.4. Estudio de las transformers

Análisis del funcionamiento de las redes *transformers*, en especial de *Vision Transformers*. Ventajas y desventajas ofrecidas por este tipo de arquitectura, identificar las diferentes alternativas dentro de las redes *transformers* y estudio de las características de cada una. Seleccionar la arquitectura que mejor solución pueda ofrecernos dentro de las alternativas identificadas. En este caso se ha seleccionado la red Restormer.

3.5. Diseño e implementación del sistema de cancelación

Con la red *Restormer* ya seleccionada se ha pasado a la implementación de esta. Para ello se han determinado las diferentes herramientas que van a ser necesarias para la consecución de este proyecto:

- Python 3.7.13
- Pytorch 1.8.1
- CUDA 10.2.89 y CuDNN 7.6.5

- Anaconda 3
- MATLAB
- TorchVision 0.9.1

Todas estas herramientas han sido instaladas en un servidor de cálculo proporcionado por el grupo de investigación TSR (Tratamiento de Señal y Radiocomunicación). En el servidor se cuenta con una mayor capacidad de procesamiento, por ello todo el entrenamiento y la generación de los conjuntos de datos (*datasets*) va a ser realizada allí.

3.6. Pruebas y validación del sistema

Tras la implementación del sistema de cancelación en el servidor, se ha procedido a la definición de diferentes escenarios de transmisión donde se va a comprobar el funcionamiento de la red seleccionada. En cada uno de ellos se entrenará la red para el escenario concreto. Las soluciones planteadas se harán de manera escalada, planteando primero soluciones sencillas que permitan aprender las características de la red para este problema, haciendo que las soluciones finales otorguen una mejor estimación debido a las mejoras introducidas a la red.

3.7. Análisis de rendimiento

Se realizará un estudio exhaustivo de los modelos generados en el apartado anterior. Realizando un análisis de resultados donde se extraerán las características y particularidades de estos. Se analizará si el sistema implementado es válido y se plantearán mejoras posibles de este sistema a futuro, así como de nuevas líneas de investigación descubiertas con estos análisis.

4. Definición del problema

El problema a abordar en este proyecto es el de la autointerferencia, por ello partiendo de la *Ecuación* (1) se ha determinado que se desea eliminar el término de loopback. De esta manera se deja la recepción como un problema tradicional de comunicaciones, donde lo que tenemos es una señal de interés, un canal de propagación y un ruido, *Ecuación* (2).

$$y = x * h + n \quad (2)$$

Para lograr obtener esta ecuación será necesario conocer el canal de loopback y la señal de loopback para poder restárselas a la *Ecuación* (1). Como la señal de loopback es conocida, debido ser la que se está transmitiendo en esa misma torre, solo es necesario obtener una buena estimación del canal de loopback para convolucionar ambos términos y restárselos a la señal recibida. Es por ello que dentro del esquema planteado en la *Figura 4* el bloque cancelador propuesto se compone de un estimador de canal, pensado para lograr estimar el canal de loopback, y un segundo bloque cancelador donde se resta esta estimación junto con la señal loopback conocida.

Para lograr estimar solamente el término h_{LB} , se pasa la *Ecuación* (1) del dominio del tiempo al dominio de la frecuencia mediante la transformada de Fourier. Con ello se obtiene la siguiente ecuación en frecuencia:

$$Y(f)_{RX} = X(f)_{FW}H(f)_{FW} + X(f)_{LB}H(f)_{LB} + N \quad (3)$$

A esta *Ecuación* (3) se la divide entre la señal X_{LB} , de esta una nueva *Ecuación* (4). Es posible realizar esta división debido a que X_{LB} es la señal que transmitirá en la torre y por ello es conocida. En la *Ecuación* (4) se observa como existen dos términos, siendo uno de ellos el canal de loopback en solitario. Gracias a esto es posible diferenciar por completo el canal de loopback del resto de la señal, facilitando así su estimación.

$$\frac{Y(f)_{RX}}{X(f)_{LB}} = \frac{X(f)_{FW}H(f)_{FW} + N}{X(f)_{LB}} + H(f)_{LB} \quad (4)$$

Esta *Ecuación* (4) es el punto de partida para la resolución de la estimación de canal. A partir de ella se tratará de buscar maximizar la componente de H_{LB} , tratando de anular por completo la otra. Para ello, como se ha explicado con anterioridad, se va a hacer uso de IA, concretamente se va a emplear una de las técnicas más desarrolladas conocida como aprendizaje supervisado. El aprendizaje supervisado consiste en entrenar la IA mediante una gran cantidad de datos que ya han sido clasificados con anterioridad. La IA recibe a su vez la pregunta, en este caso la señal en recepción dividida entre la señal de loopback, y la respuesta, el canal de loopback. Esto tiene como finalidad que, mediante el algoritmo, se combinen estas dos informaciones para crear un modelo predictivo que sea capaz de predecir la etiqueta del dato de entrada sin necesidad de introducirla.

Para poder emplear este tipo de aprendizaje con técnicas de IA será necesario contar con un gran conjunto de datos, por un lado teniendo la señal completa de recepción, y por otro el canal de loopback a estimar. Estos datos han sido generados de manera manual

mediante unos scripts haciendo uso de la herramienta software *MATLAB*. Se han creado todos los elementos que componen la señal de recepción, x_{FW} , h_{FW} , x_{LB} , h_{LB} y n . De manera que se cuenta con todas las señales de interés de manera separada, para generar la señal en recepción simplemente se ha seguido la *Ecuación* (1). El estándar de televisión utiliza comunicaciones basadas en técnicas OFDM donde el número de subportadoras por símbolo suele ser de 2^{11} . Es por ello que se han generado los datos con 2048 subportadoras, el número de instantes temporales ha sido de 20000. De esta manera, los datos generados tienen una estructura de 2048×20000 . El intervalo de guarda fijado en estas comunicaciones ha sido de $\frac{1}{8}$.

El problema se ha planteado con los datos estructurados en dos dimensiones, esto se debe a numerosas razones que se presentan a continuación:

- Una señal de telecomunicaciones se entiende de manera mucho más intuitiva mediante la representación conjunta de los instantes temporales y de las realizaciones del canal. Una señal de telecomunicación se define como la evolución temporal de las realizaciones de esta.
- Para cumplir las condiciones de las señales naturales explicadas en la sección 1 las señales deben de estar estructuradas de manera concreta. En este caso, los instantes temporales y las subportadoras deben estructurarse de manera continua, donde las subportadoras van colocadas de manera que aumenta su valor en frecuencia. Los instantes temporales se colocan uno detrás de otro debido a la característica de dependencia con el avance del tiempo de estas señales.

Debido a esta estructuración de las señales, se han estudiado técnicas IA basadas en la resolución de problemas en dos dimensiones. Esto presenta una gran ventaja debido al amplio número de soluciones existentes para problemas de 2 dimensiones. Se ha decidido estudiar las técnicas de IA de visión artificial tales como: redes convolucionales, redes LSTM (Long short-term memory) y *transformers*.

5. Análisis del estado del arte

5.1. Procesado de señal

Todo sistema de telecomunicaciones cuenta con algún tipo de interferencia o degradación de la señal de interés, es por ello que la señal obtenida en recepción no suele ser la señal deseada. Debido a esto, en cualquier sistema de telecomunicaciones es necesario implementar un procesamiento de señal que permita limpiar la señal recibida y extraer de ella la señal deseada. Este procesamiento es diferente para cada tipo de esquema de comunicaciones. En el caso de este proyecto, se han estudiado los procesamientos de señal referidos a la eliminación de autointerferencia. Para este problema existe un estado del arte muy desarrollado donde muchos algoritmos han sido creados, esto se debe al auge de los OCR (On Channel Repeater) [14] los cuales presentaban ya esta característica.

5.1.1. Ecuación

La técnica de ecualización propuesta en el documento [15] presenta una solución basada en un filtro FIR (*Finite Impulse Response*). Esta solución presenta la estructura que se muestra a continuación:

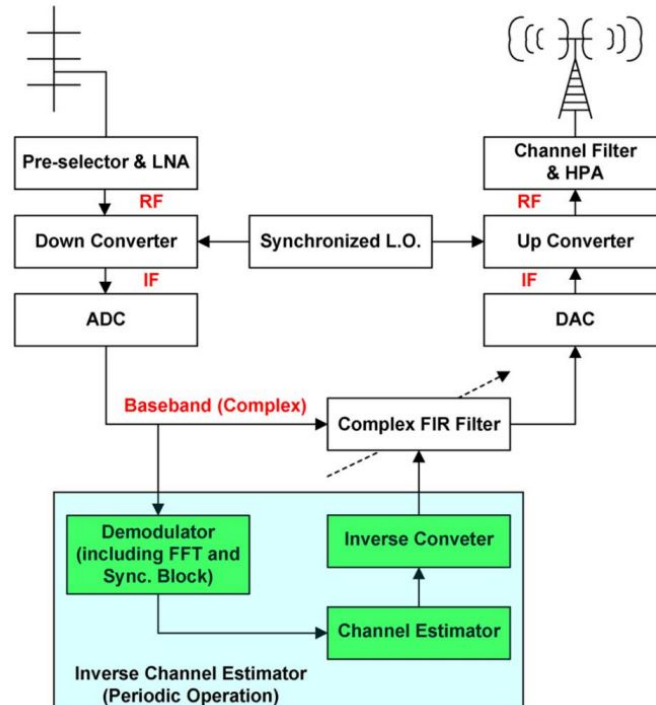


Figura 6: Técnica de ecualización para la estimación de canal

El filtro FIR cuenta con un bloque que le proporciona la señal del canal interferente. Este bloque estimador está compuesto por: un demodulador, un bloque estimador de canal y un

inversor. El canal estimado se invierte para introducirlo dentro del filtro FIR y anulando de esta manera el canal interferente. El demodulador se encarga de detectar los símbolos de referencia de fase (conocidos por sus siglas en inglés PRS). Estos PRS son señales pilotos empleadas para la estimación del canal, se insertan en los símbolos OFDM de manera conocida. El demodulador detecta estos PRS y los sincroniza para después enviárselos al estimador de canal. El estimador emplea estos símbolos PRS recibidos por el demodulador, Y_{PRS} , y los conocidos de los símbolos OFDM X_{PRS} para estimar el canal. Se realiza una comparación entre los recibidos y los enviados, obteniendo de esta diferencia la estimación de canal.

$$H_{PRS} = \frac{Y_{PRS}}{X_{PRS}} + N \quad (5)$$

N es el ruido blanco y H_{PRS} es el canal estimado con las señales piloto. Podemos observar que el canal estimado tiene un efecto de distorsión por el ruido. La técnica empleada para la estimación del canal está basada en el método LS (Least Square).

Por último, este canal se introduce en un inversor, el cual invierte el canal estimado para pasárselo al filtro FIR y que este se lo sume a la señal recibida, anulando así este canal interferente.

5.1.2. Gestión dinámica de ganancia

La técnica presentada en el informe [16] presenta un nuevo enfoque de cancelación basado en un esquema de gestión de ganancia. Este presenta un bucle de ganancia con el que se pretende mejorar la estimación del canal. El bucle utiliza una métrica de ganancia de bucle como indicador de estabilidad y calidad de la señal de salida (SNR).

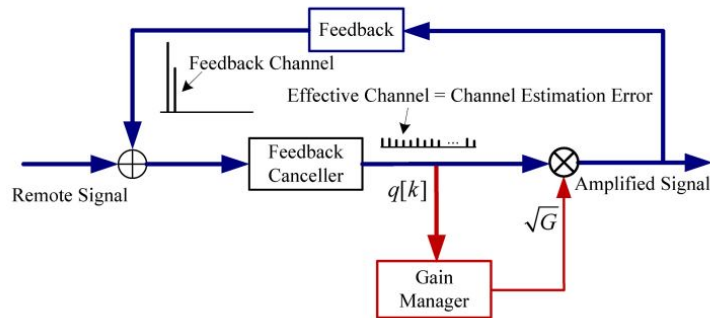


Figura 7: Bucle de ajuste dinámico de ganancia

Al igual que en el caso presentado en la sección 5.1.1 se utiliza un estimador de canal basado en una señal de referencia conocida. Es decir, se usa una señal conocida introducida en la señal enviada como parámetro para evaluar la distorsión introducida por el canal y poder así evaluar este. Sin embargo, en este esquema tras la cancelación de la señal se calcula la señal residuo q . A esta señal se le aplica una ganancia y se usa como realimentación del estimador para mejorar la estimación. La ganancia aplicada se calcula utilizando la

métrica de ganancia de bucle que hace que mejore la estimación de canal, esta ganancia es dinámica cambiando cuando se detectan cambios en la cancelación q .

5.1.3. Cancelación en dos pasos mediante filtrado digital

La última solución estudiada se presenta en el artículo [17]. Se trata de una técnica de cancelación de eco en dos etapas basada en la reconstrucción de la respuesta impulsional del canal de acoplamiento mediante filtrado digital. La estimación se consigue mediante la inyección en la señal de secuencias conocidas generadas de manera local. En este esquema

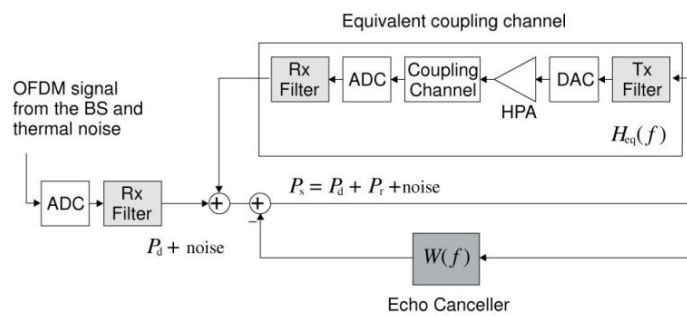


Figura 8: Esquema de cancelación en dos fases

de la *Figura 8* se observan los dos puntos de cancelación, el primero hace uso del canal estimado y el segundo de un filtro cancelador de ecos. La estimación de canal se realiza siguiendo los mismos pasos que en los métodos anteriores, con la diferencia que esta técnica hace uso de la métrica MMSE (Minimum Mean Square Error) en lugar de LS. La novedad de este método es el uso de una segunda etapa de cancelación donde se emplea el filtro cancelador de ecos. Este se encuentra realimentado por la señal de salida, cambiando así de manera dinámica en el tiempo. Los coeficientes de este filtro son calculados antes de la puesta de funcionamiento del sistema. Este cálculo se realiza mediante secuencias de entrenamiento, las cuales optimizan los coeficientes del filtro para la resolución de este problema de estimación.

5.2. Inteligencia Artificial

La gran evolución que está teniendo la IA, y que se estima que seguirá teniendo, hace que siempre sea interesante explorar estas técnicas como solución a todo tipo de problemas físicos/matemáticos. Además, la IA ha demostrado ser la mejor alternativa para la resolución de problemas que implican el procesamiento de señales naturales, tales como el reconocimiento de imágenes, el procesamiento de los lenguajes naturales... Debido a que las señales con las que se va a trabajar son señales naturales, el estudio de técnicas de IA es muy conveniente.

Por otro lado, el estado del arte presentado se centra en técnicas de IA basadas en la resolución de problemas de dos dimensiones. Esto se debe a que el planteamiento del problema de estimación puede enfocarse fácilmente como un problema en dos dimensiones, siendo una de las dimensiones los símbolos OFDM y la otra la evolución temporal de estos. Como la evolución temporal de los símbolos está correlada entre los diferentes instantes, es muy beneficioso plantear el canal como un dato en dos dimensiones. Pudiendo extraer características generales del canal en lugar de cada vector OFDM por separado. Por ello se han estudiado diferentes arquitecturas de IA para la resolución de problemas de dos dimensiones: redes convoluciones, Long Short-Term Memory y transformers.

5.2.1. Redes convolucionales

Las redes neuronales convolucionales (por sus siglas en inglés, CNN) son uno de los tipos más conocidos de redes neuronales debido a su gran aportación a las tareas de visión artificial. Estas redes fueron diseñadas para aprovechar la estructura espacial que presentan las imágenes. Con este tipo de redes se consigue relacionar la información de un píxel con los adyacentes, pudiendo así obtener no solo información del píxel, sino información más compleja formada por el conjunto de píxeles. Esta información va desde líneas y curvas hasta llegar a estructuras mucho más complejas como caras, coches o cualquier tipo de forma.

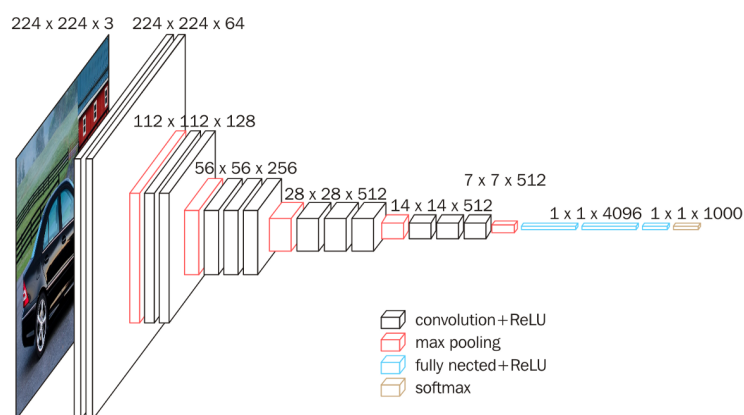


Figura 9: Arquitectura de una red convolucional

Estas redes utilizan unas operaciones conocidas como convolución para obtener la informa-

ción. Esta operación se basa en unas matrices (conocidas como filtros o *kernel*) que se aplican sobre la imagen para obtener una nueva imagen, la cual es conocida como mapa de características [18]. Estos mapas de características son los que otorgan información más compleja relacionando los píxeles vecinos. Por ejemplo, estos mapas pueden otorgar información sobre las líneas verticales que componen la imagen original, *Figura 10*. Inicialmente, los mapas de características son muy simples, solo líneas verticales u horizontales, pero aplicándoles otros filtros se consiguen mapas que contienen formas más complejas. Es por ello que estas redes cuentan con varias capas, como se puede observar en la *Figura 9*. En esta *Figura 9* también se puede observar que a medida que se avanza en la red las imágenes generadas serán más pequeñas, pero contaremos con un mayor número de ellas. Esto se debe a que a medida que se avanza los píxeles de las imágenes ya contienen la información espacial de sus vecinos, por ello el tamaño de las imágenes se reduce. Del mismo modo, debido a la complejidad de los objetos a detectar, el número de mapas de características va en aumento.

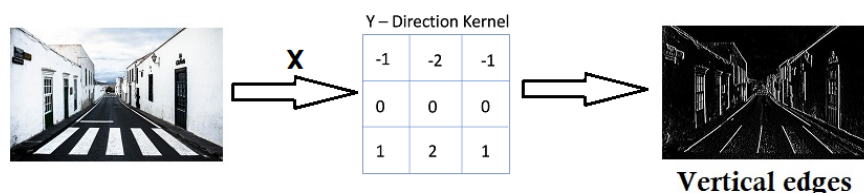


Figura 10: Operación de convolución para la obtención de un mapa de líneas verticales

El paso final en las redes convolucionales suele ser la clasificación de la imagen en diferentes etiquetas, sin embargo, este no es el objetivo de este proyecto. En las redes convolucionales tradicionales se utilizan perceptrones multicapa para esta clasificación, en este caso este último elemento no será necesario. Para este proyecto se necesitará combinar la información de estos mapas de características para poder generar la señal deseada. Esto se consigue mediante otra capa convolucional que combine estos mapas de características para conseguir la imagen deseada.

Durante el entrenamiento de estas redes son los valores de los filtros lo único que se modifica. Mediante el empleo de técnicas de descenso de gradiente, se van ajustando estos valores hasta obtener los mapas de características que optimicen nuestro problema. Esta optimización se evalúa mediante la comparación de la imagen generada y la imagen objetivo introducida en la red, esto es debido al uso de aprendizaje supervisado. Esta comparación se aplica, en cada uno de los ciclos del entrenamiento, a una función de error seleccionada, buscando en ella mediante los mecanismos de descenso de gradiente mencionados el punto que minimice esa función de error.

Debido al gran número de arquitecturas de redes convolucionales existentes, se ha realizado un estudio de las diferentes subfamilias de estas.

Autoencoders

Las redes conocidas como autoencoders son un caso particular de las CNNs donde no solo se extraen las características de una imagen, sino que se hace uso de estas para la generación de nuevas imágenes sin alguna de las características indeseadas que puede presentar esta. Estas redes están compuestas por una estructura en espejo, como se puede apreciar en la *Figura 11*, donde las capas convolucionales que generan los vectores de características de la imagen luego se invierten para poder generar estas imágenes. Esta arquitectura en su inicio está formada por las capas convolucionales tradicionales de una CNN, tras estas capas la información relevante de la imagen ha sido extraída. Las capas finales de la red son capas deconvolucionales, estas son las encargadas de, a partir de información extraída, volver a muestrear estos mapas de características logrando la nueva imagen. La etapa donde se extraen las características de la imagen es conocida como *Encoder* y la encargada de la reconstrucción es conocida como *Decoder*.

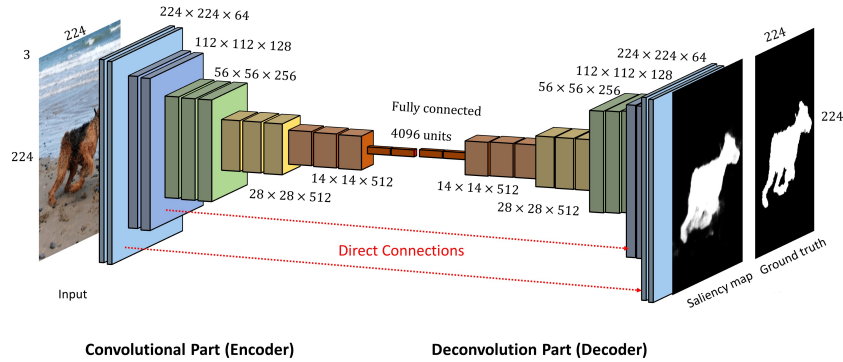


Figura 11: Arquitectura de una red CNN autoencoder

Estas redes son principalmente empleadas para tareas eliminación de ruido en imágenes (‘Image Denoising’) [19]. En este tipo de tareas el problema se modela como una imagen X , la cual tiene una función que la deteriora H y un ruido aditivo n como se muestra en la siguiente ecuación:

$$Y = H(x) + n \tag{6}$$

Generalmente, el entrenamiento de estas redes se realiza con un conjunto de imágenes tradicional al cual se le añade un ruido conocido. Los datos de entrada son las imágenes con el ruido añadido y las etiquetas (*label*) con las que se compara son las imágenes normales. De este modo, estas redes pueden obtener la información asociada a dicho ruido y eliminarlo cuando se reconstruye la imagen en el *decoder*. Es decir, aprenden a identificar el parámetro n de la *Ecuación* (6) y generan una nueva imagen sin él. Debido a este proceso de aprendizaje explicado y observando las similitudes entre las ecuaciones (4) y (6) se puede concluir que este tipo de redes podrían ser empleadas para la resolución del problema planteado en este proyecto. Para ello es necesario un cambio en el método de entrenamiento de estas redes, donde se elimine la adición del ruido y donde el parámetro a eliminar de la *Ecuación* (4) sea el que contiene todas las señales menos el canal de loopback.

SuperResolution

La super resolución de imágenes es otro de los problemas de visión por ordenador más extendidos. Este problema consiste en la partida de una imagen con una baja resolución para terminar consiguiendo esa misma imagen pero con una mayor resolución. En los últimos años, se han comenzado a emplear redes convolucionales como una de las nuevas alternativas a la solución de estos problemas. Una de las arquitecturas seleccionadas para la resolución de este problema es la *Super-Resolution Convolutional Neural Network* presentada en [20]. Esta red es una red de 3 capas (*Figura 12*), por lo que la hace una red muy simple y con poco coste computacional. Esta red se basa en la ampliación de patrones detectados dentro de la imagen, de manera que de unos píxeles de información es capaz de crear un mayor número de píxeles que ilustren esa información adquirida. De esta manera, la red se encarga de potenciar la información embebida dentro de una imagen, es por ello que se ha pensado como una posible solución al problema planteado en el proyecto. La red sería capaz de obtener y potenciar información oculta (el canal de loopback), dentro de toda la información existente en la imagen.

En este tipo de redes se parte de una imagen original de baja resolución Z , a esta imagen se le aplica una interpolación lineal para aumentar su resolución, creando así una nueva imagen Y . Esta imagen Y cuenta con la misma resolución que la imagen salida de la red $F(Y)$. Esta imagen $F(Y)$ deberá de parecerse lo máximo a la imagen usada como etiqueta, conocida como X . Esta red puede emplearse para la estimación de canal prescindiendo de la parte de interpolación, debido a que en nuestro problema no es necesario el aumento de la resolución de la imagen de entrada.

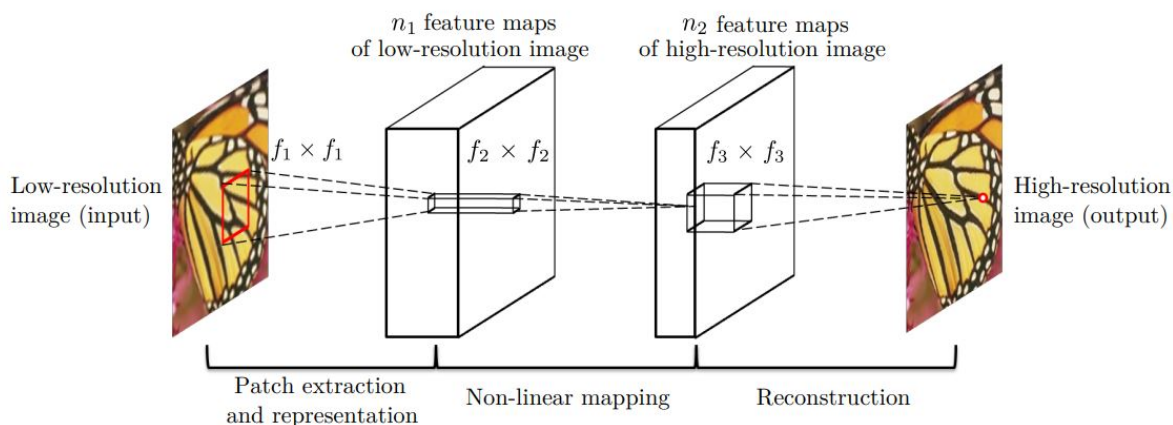


Figura 12: Arquitectura de la red SRCNN

Como ya se ha mencionado, esta red cuenta con 3 capas, las cuales tienen funciones diferenciadas. Siendo la primera la encargada de la extracción y representación de las secciones de la imagen, la segunda de mapeo no lineal y la tercera de la reconstrucción.

- Extracción y representación de secciones:** Esta capa se encarga de extraer secciones de la imagen de baja resolución Y y representa cada sección como un vector de grandes dimensiones. Esta capa se encarga de comprimir mapas de características en estos vectores, el número de vectores dependerá del tamaño de filtro utilizado. En esta capa se convoluciona la imagen con los filtros W_1 y se les suma un valor B_1 . La

expresión de esta capa queda como:

$$F_1(\mathbf{Y}) = \max(0, \mathbf{W}_1 * \mathbf{Y} + \mathbf{B}_1) \quad (7)$$

El tamaño de los filtros W_1 se define como $f_1 \times f_1$, siendo este f_1 la dimensión espacial de la región que se desea representar. El valor W_1 representa tanto el tamaño de los filtros empleados, como el número de filtros n_1 . Este número n_1 determina el número de vectores generados.

- **Mapeo no lineal:** Esta capa se encarga de mapear cada uno de los n_1 vectores de baja resolución, generados en la capa 1, a los n_2 vectores de alta resolución que son la salida de la capa 2. Este mapeo no lineal no se hace sobre la información de la imagen, sino que se realizan sobre los mapas de características que se han obtenido en la capa 1, es por ello que la operación realizada en la segunda capa queda definida como:

$$F_2(\mathbf{Y}) = \max(0, \mathbf{W}_2 * \mathbf{F}_1(\mathbf{Y}) + \mathbf{B}_2) \quad (8)$$

W_2 contiene los n_2 filtros de tamaño $f_2 \times f_2 \times n_1$. Estos n_2 vectores son la representación conceptual de las secciones de la imagen de alta resolución.

- **Reconstrucción:** La última capa de la red es la encargada de generar la imagen de alta resolución a partir de las secciones de alta resolución generadas en la capa anterior. Estas secciones muchas veces suelen ser promediadas para la generación de la imagen final. Por tanto, esta capa queda definida como:

$$F(\mathbf{Y}) = \mathbf{W}_3 * \mathbf{F}_2(\mathbf{Y}) + \mathbf{B}_3 \quad (9)$$

En este caso W_3 corresponde con los c filtros de tamaño $f_3 \times f_3 \times n_2$. Siendo c el número de canales de la imagen, en nuestro caso 1. Los filtros aplicados en esta capa son todos lineales.

A pesar de que las funciones que realiza cada capa son diferentes, todas hacen uso de capas convolucionales para llevarlas a cabo. En este modelo de red los valores optimizados durante el entrenamiento de la red serán: los pesos de los filtros y los valores bias (B_1 , B_2 y B_3).

VGG16

Se trata de una de las redes convolucionales más famosas, emplea una arquitectura más compleja que la mayoría de las demás CNNs. Es una de los mejores modelos para las tareas de clasificación mediante visión artificial [21]. Es una red que cuenta con 16 capas convolucionales, donde los filtros convolucionales empleados son pequeños, de dimensión 3×3 [22]. Gracias al uso de filtros de menor tamaño se aumenta la profundidad de la red, haciendo que de este modo mejore significativamente el rendimiento. Con este uso de filtros pequeños se logra que los mapas de características generados a partir de la imagen mantengan una mayor correlación espacial. Además, con el uso de capas convolucionales de filtro de tamaño pequeño conectadas en cascada, se consigue reducir de manera significativa el número de parámetros de la red. El número de parámetros N introducidos por una capa se calcula como:

$$N = f_1^2 \times C^2 \quad (10)$$

Siendo f_1 el tamaño del filtro y C el número de canales de la imagen. Por tanto, si se usa una capa con filtros de tamaño 7 el número de parámetros N ascenderá a $49 \times C^2$ mientras que con el uso de 3 capas de filtros de tamaño 3, el total de parámetros N será de $3 \times 9 \times C^2$, es decir, a $27 \times C^2$. Gracias al uso de filtros más pequeños no solo aumenta la eficiencia de la red, sino que se reduce también el número de parámetros de esta. Por último, el uso de estos filtros también mejora el rendimiento de la red debido a la aplicación de más de una función de activación, haciendo que esto sea todavía más discriminativo.

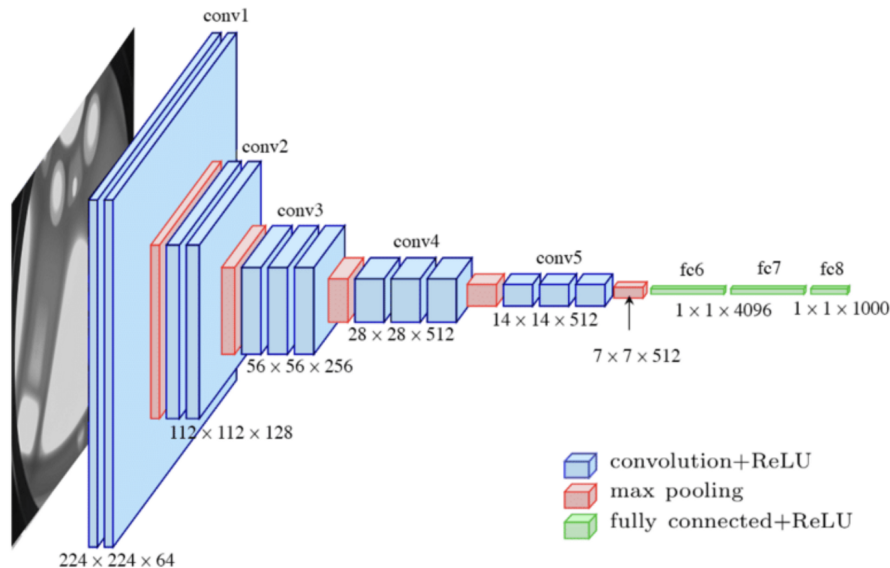


Figura 13: Arquitectura de la red VGG16 net

En esta red cada ciertas capas se usan capas de reducción conocidas como *Max pooling*. Estas capas se encargan de la reducción del tamaño de los mapas de características generados, de esta manera se reducen el número de parámetros. A pesar de todo ello, debido a su complejidad, esta red cuenta con un gran número de parámetros (alrededor de 133 millones). Esta gran cantidad de parámetros hace que su implementación pueda llegar a ser mucho más costosa. Además, al ser una red pensada para tareas de clasificación,

se deberá adaptar la arquitectura para poder emplearlo en el problema de este proyecto. Por último, debido al tamaño fijo de las primeras capas de la red, también se deberán de adaptar los datos de entrada a esta red.

ConvNext

Se trata de un novedoso tipo de red surgido en el año 2020. Se trata de redes diseñadas para poder competir con las *Visual Transformers (ViT)*, las cuales son las redes que mayor rendimiento están presentando actualmente para las tareas de visión artificial. Las ViT nacieron como una adaptación de las redes Transformers para problemas de visión. Desde su diseño han tenido un gran impacto en este tipo de tareas, desbancando de la primera posición a las CNNs como solución a los problemas de visión artificial. Debido al rendimiento histórico de las CNNs y a la hipótesis de que la convolución sigue siendo la mejor herramienta para las tareas de visión artificial, varios investigadores decidieron seguir desarrollando las redes convolucionales, pero aplicando varias de las técnicas de las *transformers*. Tratando así de competir con estas para poder volver a ser las redes dominantes en las tareas de visión artificial [23].

El cambio de estructura de una red convolucional (ConvNet) no ha sido algo inmediato, sino que se han seguido varios pasos para llegar hasta el concepto final de estas redes ConvNext.

- **Técnicas de entrenamiento:** Las técnicas de entrenamiento empleadas no eran óptimas y siguiendo el camino marcado por las ViT se ha descubierto que aumentando el número de *epoch* de una red tradicional como la ResNet-50 [24] el rendimiento de esta mejora. También se ha decidido comenzar a usar otro tipo diferente de optimizadores, técnicas conocidas como *data augmentation*.
- **Macro diseño:** Se han introducido cambios en el diseño de las capas de las redes haciendo uso de lo aprendido de las ViT. Las ViT usan técnicas conocidas como *patchify*, es decir, dividen la imagen de partida en imágenes más pequeñas, las cuales luego relacionan. Inicialmente en las CNN se reduce el tamaño de la imagen a vectores de características más pequeños, esto se hace igual en las ViT. Sin embargo, las ViT lo hacen de una manera más agresiva debido a la reducción ya realizada con la división de la imagen. Por ello las nuevas ConvNext harán uso de una nueva capa de red conocida como *patchify* para implementar estas mejoras.
- **Correlación espacial y de canal:** Las ViT son capaces de relacionar la información de cada uno de los canales de la imagen (los colores R, G y B) con los demás. Consiguiendo así una mayor información del contenido espacial de la imagen mediante la relación de ellos. Para emular también este comportamiento en las CNN se ha decidido el uso de un tipo especial de convolución conocido como *Depthwise convolution*, la cual relaciona la información de los 3 canales.
- **Introducción del *Inverted bottleneck*:** También se ha añadido el concepto de cuello de botella invertido existente en las ViT. Este concepto consiste en llevar la capa con menor número de parámetros al inicio del bloque. De esta manera se reduce el número de operaciones que tiene que realizar la red.

- Filtros de mayor tamaño:** A pesar de lo establecido por la red VGGNet de usar capas de filtros 3x3 en cascada, las ViT han demostrado, con sus métodos de atención, que es mucho más beneficioso tener un mayor contexto de la imagen analizada. Por ello, el uso de filtros de mayor tamaño es recomendado, sin embargo, al pasar de filtros 7x7 la mejoría se satura.
- Cambios menores:** Por último, se han estudiado los cambios en los elementos de la micro arquitectura de las redes. Empezando por un cambio en las funciones de activación empleadas al final de las capas, se ha cambiado la tradicional ReLU (Rectified Linear Unit) por GELU (Gaussian Error Linear Unit). Se ha imitado a las *transformers* empleando un menor número de funciones de activación. También se reducen el número de capas de normalización, y el tipo de capa de normalización.

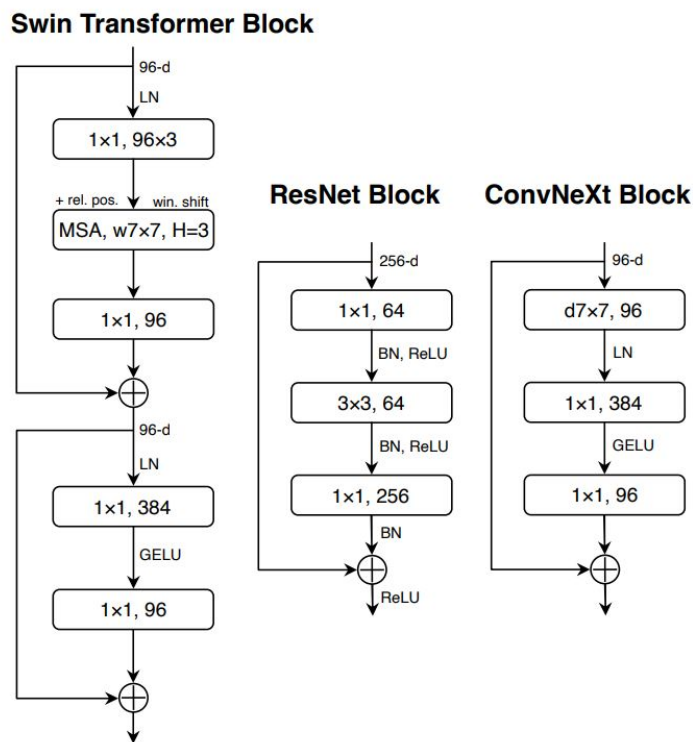


Figura 14: Comparación entre los bloques de una transformer, una CNN (ResNet) y una red ConvNext

5.2.2. LSTM

LSTM (Long Short-Term Memory) es un modelo de red basado en las redes neuronales recurrentes (RNN). Fueron diseñadas para hacer frente al problema de desvanecimiento del gradiente, el cual aparece cuando los datos tienen dependencias a largo plazo. Es por ello que estas redes se pensaron para problemas variables en el tiempo, donde las variaciones temporales se encuentran correladas entre sí. Estas redes se caracterizan por estar construidas mediante celdas. Estas celdas son redes recurrentes a las cuales se les aplica una capa de entrada (*Input gate*), una capa de salida (*Output gate*) y una capa conocida como *forget gate* [25]. Estos tres elementos conforman lo que se conoce como bloque LSTM, el cual es conocido como celda de memoria (*Figura 15*). La arquitectura de una red LSTM se forma a partir de la conexión de estos bloques. Son los encargados de mantener el estado a lo largo del tiempo y modificar esta con el flujo de información que se transmite por la red. Estas redes han sido empleadas en problemas que requieran de memoria temporal. Se han usado para la predicción de series temporales, donde el objetivo es, basándose en comportamientos pasados, poder predecir tendencias futuras. Por ejemplo, se han realizado modelos de predicción del mercado de finanzas. También han sido empleadas en tareas de procesamiento natural del lenguaje (NLP), debido a su capacidad de capturar información a largo plazo existente en las frases del lenguaje humano.

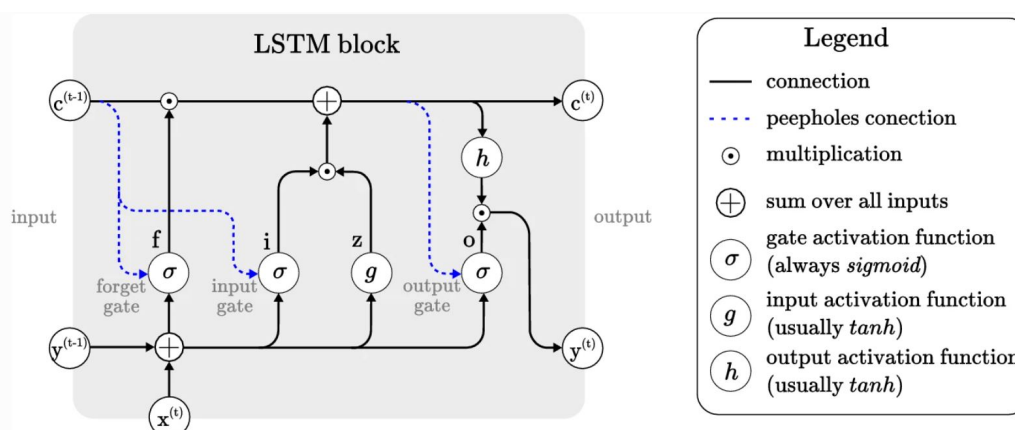


Figura 15: Arquitectura de un bloque LSTM

El funcionamiento de las celdas se divide en diferentes pasos, los cuales se realizan de manera secuencial. En estos se van utilizando las diferentes capas observadas en la *Figura 15*. Para entender el funcionamiento del bloque LSTM hay que distinguir entre dos conceptos, el primero, la información de estado de la celda C^t y la salida de la celda y^t . El estado de la celda C^t es el encargado del mantenimiento de la información de instantes pasados, se va actualizando en cada ciclo de operaciones de la celda. Los pasos seguidos en estos bloques son:

- **Forget gate:** Es la primera operación realizada en el bloque, se encarga de determinar que información del estado anterior de la celda C^{t-1} se debe eliminar. Para ello genera el valor de activación f . Este valor se calcula haciendo uso de la entrada

actual x^t , la salida y el estado de la celda en el instante anterior (y^{t-1} y C^{t-1}) y el término de bias b_f . A todo esto se le aplica la función de activación sigmoid, σ

$$f_t = \sigma(W_f x^t + R_f y^{t-1} + p_f c^{t-1} + b_f) \quad (11)$$

Donde los valores, W_f , R_f y p_f corresponden a los pesos de la red asociados a las señales x^t , y^{t-1} y C^{t-1} respectivamente.

- **Input gate:** Es el paso donde se actualiza el valor de entrada en la celda mediante la combinación de la entrada actual x^t y la salida y estado de la celda en la iteración previa. Así se genera el vector i^t el cual representa el valor de entrada de la red. Se utiliza como función de activación sigmoid, σ

$$i_t = \sigma(W_i x^t + R_i y^{t-1} + p_i c^{t-1} + b_i) \quad (12)$$

Donde los valores, W_i , R_i y p_i corresponden a los pesos de la red asociados a las señales x^t , y^{t-1} y C^{t-1} respectivamente.

- **Block input:** El bloque encargado de actualizar el valor de entrada de la red, para ello relaciona la información de la entrada actual con la de salida de la red en la iteración previa. Se aplica una función de activación, la cual normalmente es la tangente hiperbólica, \tanh .

$$z_t = \tanh(W_z x^t + R_z y^{t-1} + b_z) \quad (13)$$

Los valores, W_z y R_z corresponden a los pesos de la red asociados a las señales x^t e y^{t-1} respectivamente.

- **Estado de la celda:** Es una combinación de los vectores adquiridos en los pasos previos. Se combina la información del estado anterior C^{t-1} con el vector f , obteniendo así la información que permanecerá en la celda. A esta se le suma la combinación del vector de entrada i y del vector de bloque b .

$$C_t = z^t \times i^t + C^{t-1} \times f^t \quad (14)$$

- **Output gate:** Se trata de la capa de salida de la celda, aquí se genera el vector o el cual será empleado para la generación de la salida del bloque y^t .

$$o_t = \sigma(W_o x^t + R_o y^{t-1} + p_o c^t + b_o) \quad (15)$$

Donde los valores, W_o , R_o y p_o corresponden a los pesos de la red asociados a las señales x^t , y^{t-1} y C^t respectivamente.

- **Salida del bloque LSTM:** Es una combinación del estado de la celda en la iteración actual y del vector de salida de la celda o .

$$y_t = \tanh(C^t \times o^t) \quad (16)$$

Como se observa, estos bloques LSTM pueden mantener información temporal sobre los datos de entrada de la red. Esto hace que estas redes se hayan considerado como una buena alternativa para trabajar con datos de canales de telecomunicaciones, como se pretende en este proyecto. Además, estas redes se han adaptado para problemas de visión artificial, lo cual las mejora para el planteamiento del problema realizado. Sin embargo, la complejidad matemática y el alto coste computacional de estas redes hace que sea interesante el estudio de otra de las arquitecturas más novedosas.

5.2.3. Transformers

Las redes conocidas como *Transformers* nacen para solucionar los problemas presentes en las Redes Neuronales Recurrentes (RNN). Estas redes RNN tienen como característica el procesamiento secuencial de la información, conectando en cadena diferentes bloques de redes neuronales cuya salida se conecta a la entrada del siguiente bloque [26]. De esta manera, la información obtenida en el bloque n se emplea en el siguiente bloque $n + 1$ consiguiendo relacionar la información de estados anteriores, igual que en las redes LSTM (sección 5.2.2). Este tipo de arquitectura recurrente presenta numerosas ventajas, pero cuenta con una gran desventaja: a medida que se va introduciendo más información, las relaciones creadas en los primeros bloques se van debilitando, debido al paso por numerosos bloques donde se generan nuevas relaciones. Esto implica que este tipo de redes no sea capaz de relacionar información extendida en el tiempo.

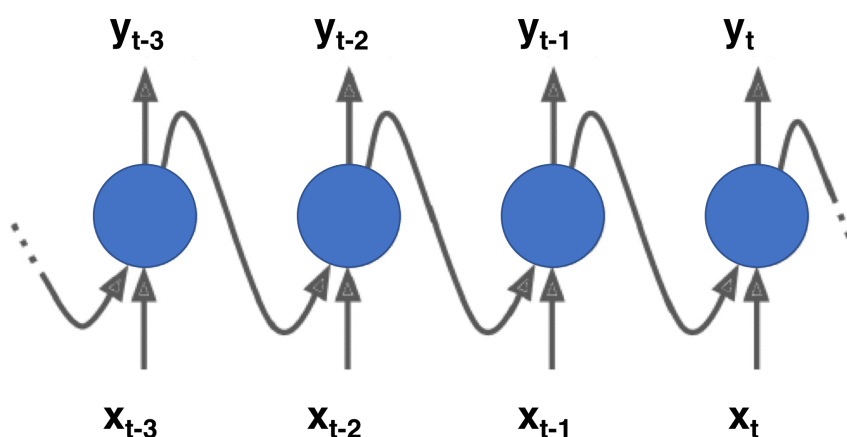


Figura 16: Arquitectura de una red RNN

Para solucionar este problema surgen los mecanismos de atención [27]. Estos mecanismos son la base del funcionamiento de las *Transformers*. Se basan en la generación de un vector conocido como vector de atención, que se generan mediante el producto escalar de dos vectores, los cuales son el vector *query* q y el vector *key* k . El vector *query* es un vector generado a través del dato de entrada, indica las características con las que pretende relacionarse este. Poniendo un caso de uso concreto, en la resolución de problemas del procesamiento del lenguaje natural (NLP), este vector representa las relaciones con otras palabras que se desean obtener de la palabra introducida como dato. El vector *key* representa las características concretas del dato de entrada. Volviendo al caso de NLP, este vector indica las particularidades de la palabra (si es un verbo, género de la palabra...). De este modo, cada uno de los datos de entrada se descompone en estos dos vectores, se realiza el producto escalar de cada uno de los vectores q de cada dato con los valores k de cada una de las palabras, obteniendo de este modo el vector de atención. Este vector es la representación de la relación existente entre los datos e indica la importancia que hay que otorgar a estas relaciones.

En la *Figura 17* observamos el valor del vector de atención de la palabra *it* con el resto de la frase. Este vector es mayor cuanto más oscuro sea el color de su representación. Este vector

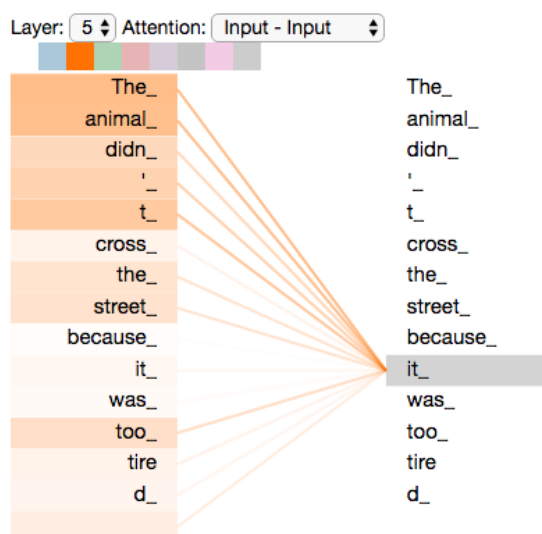


Figura 17: Mapa de atención de un problema de NLP

no es suficiente para generar el mecanismo de atención, es por ello que se genera un tercer vector conocido como *value* v . Este vector es el encargado de generar el peso que se le da a cada uno de los datos. El producto de este valor v de cada dato con la relación obtenida entre los datos será lo que finalmente se conocerá como atención, el cual representa la salida del bloque de la red donde obtenemos el dato de entrada contextualizado con el resto de datos de entrada. Expresándolo matemáticamente, este mecanismo de atención se representa así:

$$\text{Atencion}(q, k, v) = \frac{qk}{\sqrt{d_k}} \times v \quad (17)$$

Donde $\sqrt{d_k}$ es un valor de escalado que se calcula como la dimensión de datos *key* existentes.

Con este concepto de atención y la estructuración de la red neuronal en bloques de *encoder* y *decoder*, como explica en la sección 5.2.1, se crean el nuevo tipo de redes neuronales conocido como *Transformes*. Este tipo de redes se diseñaron originalmente para tareas de NLP, con una arquitectura como la que se muestra a continuación.

En la *Figura 18* se distinguen dos bloques, el de la izquierda, el cual es el *encoder* y el de la derecha, el *decoder*. Ambos cuentan con capas de atención y más capas características de las redes neuronales, siendo estas las tradicionales *feed forward*, las cuales son las redes neuronales tradicionales, y las *add and normalization*, son las encargadas de normalizar las salidas ya procesadas y añadir los datos y sus relaciones. El *encoder* recibe el dato de entrada y extrae parte de sus características, y el *decoder* recibe el dato etiquetado y trata de ajustar el dato de salida del *encoder* a este. Sin embargo, la operación más importante de estas redes se encuentra al inicio de los bloques, esta se conoce como codificación posicional. Esta operación es necesaria debido a la eliminación del procesamiento secuencial de los datos, en estas redes los datos no se introducen de 1 en 1 sino que se introduce de manera paralela un gran conjunto de datos. Debido a esto, la posición que ocupan los datos se desconocerá, la cual característica muy importante para poder obtener las relaciones entre estos. Para solucionar este problema se usa esta bloque de codificación posicional. La

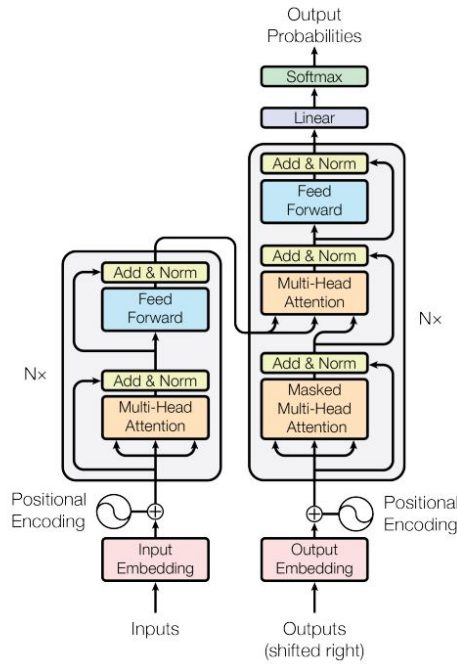


Figura 18: Bloques de una red transformer

codificación posicional consiste en añadir a los datos información que haga comprensible a la red la posición que estos ocupan dentro del conjunto de datos. Para ello se define una función cuyo valor depende de la posición y de la dimensión del conjunto de datos i .

$$PE_{pos,i} = \sin\left(\frac{pos}{i}\right) \tag{18}$$

Esta Ecuación (18) puede optimizarse como se observa en [27]. Sin embargo, el concepto detrás de esta técnica es este, siendo pos la posición en el conjunto de datos de la entrada e i la dimensión de este conjunto.

Con este funcionamiento explicado se observa como las redes *Transformers* son las mejores alternativas para trabajar con datos que contengan relación a lo largo del tiempo. Sin embargo, estas redes se emplean en soluciones de problemas de procesamiento del lenguaje natural y otros problemas donde los datos de entrada son vectores. Por lo que puede parecer que estas redes no sean útiles para el problema de estimación de canal presentado en este proyecto. No obstante, la creación de un nuevo tipo de *transformers* las cuales trabajan en tareas de visión artificial, hacen que estas redes puedan ser empleadas. Este nuevo tipo de *transformers* es conocido como *Vision Transformer (ViT)*

Vision Transformers

Las *Vision Transformers (ViT)* son un caso particular de las transformers, generado por el buen rendimiento de estas con los conjuntos de datos que presentaban información a largo plazo. Debido a que esta característica se encuentra presente en los problemas de visión artificial, se decidió crear este nuevo tipo de redes. Las *vision transformers* presentan la misma estructura de *encoder* y *decoder*, así como las mismas capas de atención que las *transformers*, sin embargo, estas redes cuentan con capas deconvolucionales para poder crear las imágenes una vez se ha realizado el aprendizaje. Al tratarse de imágenes como datos de entrada, estas se dividen en parches de menor tamaño, los cuales se enumeran al igual que una secuencia de datos tradicional. A estos parches se les aplica la codificación posicional y se les convierte en vectores de una dimensión [28]. Este último paso dependerá de la ViT empleada, existiendo ViT que usan los datos como matrices en lugar de vectores debido a la inserción de capas convolucionales en los *encoder* y *decoder*.

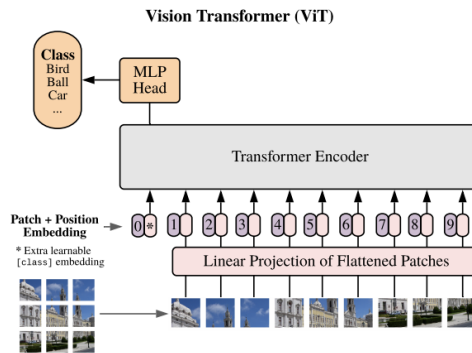


Figura 19: Arquitectura de una ViT

6. Estudio de alternativas de IA para cancelación

Tras la decisión de estudiar técnicas de IA para solucionar el problema de cancelación de canal en estos casos de autointerferencia, se ha procedido a realizar un análisis de las alternativas estudiadas basándonos en los siguientes aspectos:

- **Facilidad de implementación:** Necesidad de recursos hardware y software para poder utilizar dicha solución.
- **Coste computacional:** Coste presentado por la red, tiene en cuenta el número de parámetros de esta, así como el tiempo de entrenamiento.
- **Información a largo plazo:** Capacidad de la red de capturar información a largo plazo.
- **Flexibilidad de los datos de entrada:** Hace referencia a la necesidad de emplear conjuntos de datos con una estructura fija.
- **Necesidad de cambios en la arquitectura:** Necesidad de modificación de la arquitectura de la red, cambiar, añadir o eliminar capas de estas.
- **Códigos para problemas similares:** Antecedentes de uso de estas redes para problemas de *Image Denoising* o similares.

Tabla 1: Ventajas y desventajas de las técnicas de IA

Familias de IA	Facilidad de implementación	Coste computacional	Información a largo plazo	Flexibilidad de los datos de entrada	Necesidad de cambios en la arquitectura	Códigos para problemas similares
Autoencoder	****	**	*	****	No	Si
SRCNN	*****	*	*	****	No	Si
VGG16	***	**	*	**	Si	No
ConvNext	***	***	***	***	No	Si
LSTM	**	*****	***	***	Si	No
Transformers	**	****	*****	***	Si	No
Vision Transformers	**	****	*****	****	No	Si

Analizando estos aspectos se ha decidido utilizar las redes ViT como solución a nuestro problema, debido a ser la red que presenta mayor número de ventajas. En el siguiente apartado se presenta la arquitectura concreta de la red seleccionada.

7. Transformers como solución al problema

Las transformers son una de las redes neuronales más extendidas hoy en día. Con ellas se han creado varias de las aplicaciones más innovadoras de los últimos años, como por ejemplo:

- **GPT-3:** Modelo de procesamiento del lenguaje natural capaz de responder a preguntas como una persona humana, además de poder realizar traducciones en tiempo real y otras muchas aplicaciones relacionadas.
- **AlphaFold2:** Modelo de red que permite realizar predicciones de la estructura de las proteínas. Ha sido empleado para el descubrimiento de la estructura de las proteínas de virus como el SARS-CoV-2.
- **DALL-E 2:** Es un modelo de red que genera imágenes realistas a partir de cualquier texto introducido.

Por esta potencia de las redes y debido al análisis, se ha decidido estudiar estas como solución al problema de cancelación. Para ello, debido a la naturaleza bidimensional de los canales de telecomunicaciones, se han seleccionado soluciones basadas en *Vision Transformers*. Concretamente, se han buscado ViT empleadas para tareas de *Image denoising*, encontrando varias arquitecturas empleadas para estas tareas. El nombre de las arquitecturas encontradas y estudiadas son: *U-former* [29], *SADnet* [30] y *Restormer* [31].

De estas tres arquitecturas seleccionadas finalmente, se ha optado por elegir la red *Restormer*, debido a la mejora que presenta en las tareas de *image denoising* además de las múltiples configuraciones que permite.

7.1. Restormer

La red *Restormer* seleccionada presenta todas las ventajas estudiadas en la sección 6, además esta red añade mejoras como, una mejor captura de las interacciones entre píxeles de las imágenes de gran escala. Además, esta red es adecuada para problemas de imágenes de muchos píxeles, como es el caso de nuestro problema, debido a que los canales no son imágenes, sino conjuntos extensos de datos que serán segmentados en imágenes. Por ello, el hecho de que esta red funcione con este tipo de imágenes hace que sea una solución óptima. Presenta una arquitectura donde el *encoder* cuenta con 4 bloques, y el *decoder* está compuesto a su vez por 4 bloques también [31]. Cada bloque está a su vez compuesto por dos bloques característicos de esta red *Restormer*, estos son el *Multi-Dconv Head Transposed Attention (MDTA)* y *Gated-Dconv Feed-Forward Network (GDFN)*. El bloque MDTA es el encargado de extraer las relaciones *query* y *key* de los datos. Para ello utiliza capas convolucionales y deconvolucionales generando los vectores k , q y v . Con estos vectores genera un mapa de características ya con la atención aplicada, donde los píxeles ya se encuentran generados con el contexto de sus vecinos. Tras ello aplica el GDFN, el cual es el encargado de transformar las imágenes y de que esta información se transmita a lo largo de la red.

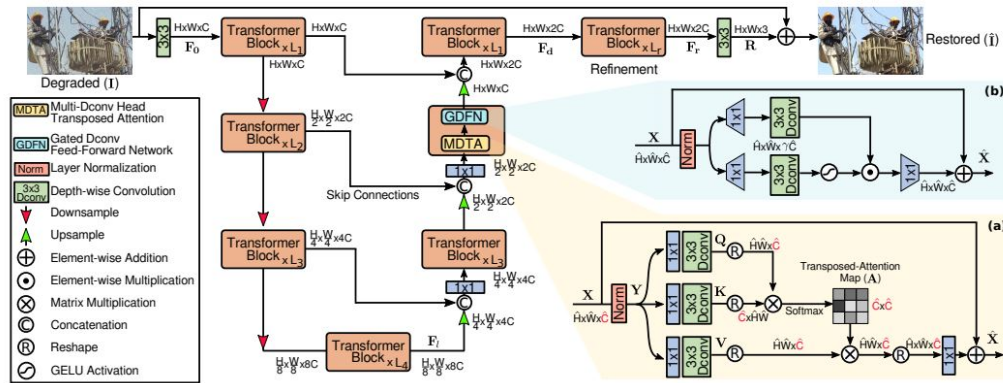


Figura 20: Bloques de la red *Restormer* [31]

En el *encoder* tras la aplicación del bloque se realiza una reducción del tamaño de la imagen, esto es similar al proceso realizado en una red convolucional. De esta manera, se aumentan las relaciones entre píxeles más alejados, además de reducir el coste computacional del entrenamiento. Esta reducción hace que a la salida de un bloque la imagen tenga la mitad de resolución espacial que a la entrada. Al llegar a la salida del último bloque, esta imagen tendrá un octavo de la resolución espacial inicial. Por ello, los datos de entrada deberán ser matrices con anchura y largura múltiplo de 8. El *decoder* realizará el proceso inverso, pasando de estas imágenes de menor resolución a la resolución de la imagen original. Los bloques del *decoder* y *encoder* se conectan, relacionando los que trabajan con las mismas resoluciones.

Esta red tiene múltiples configuraciones, siendo de interés la configuración de *real Denoising*, ya que es la que se asemeja más al problema de cancelación de canal. Estas configuraciones no presentan una gran variación en la arquitectura de la red, pero sí ayudan a converger a un buen modelo predictivo de manera más rápida. Entrando ya en características más técnicas, esta red es una red compleja que cuenta con un total de 26.109.076 parámetros a entrenar.

8. Pruebas y validación

8.1. Despliegue y configuración de la plataforma SW

Como ya se ha mencionado en la sección 3.5 se va a trabajar sobre un servidor de cálculo, es por ello que en primer lugar, se debe crear un entorno virtual para poder trabajar con la inteligencia artificial. Se define como: un espacio independiente de la instalación local del servidor, con el objetivo de aislar los recursos y librerías. Las razones por las que se debe usar un entorno virtual son:

- **Organización:** es conveniente tener controladas las librerías y las versiones de las herramientas con las que se está trabajando para que no existan incompatibilidades. Debido a esto es necesario organizar todo en entornos virtuales, ya que se aislará el proyecto que se esté realizando de otros proyectos donde las versiones o librerías utilizadas sean diferentes.
- **Estabilidad:** el hecho de que las librerías cambien de versión muy a menudo puede provocar que, si se desea llevar el proyecto a otras máquinas, estas utilicen versiones diferentes, lo cual hará que el proyecto no funcione. Esto se resuelve implementando un entorno virtual. Además de este modo se evita la actualización automática indeseada que podría provocar un funcionamiento erróneo en el código.
- **Trabajar con diferentes versiones y librerías de python:** ya que las aplicaciones utilizan versiones de python diferentes según las tareas a resolver, es muy útil trabajar con entornos virtuales debido a que se podrán emplear diferentes versiones de python para cada entorno, incluso dentro de un mismo servidor.

Para la creación de este entorno virtual se ha empleado el gestor de entornos virtuales *Anaconda*. Todas las herramientas mencionadas en la sección 3.5 han sido instaladas en este entorno. Tras la instalación de estas herramientas ha sido necesaria la configuración de los controladores de las tarjetas gráficas (GPUs) del servidor para que estas sean compatibles con la versión de *CUDA* instalada. *CUDA* es una plataforma de computación paralela que permite el uso de tarjetas gráficas. Gracias a estas es posible ejecutar el código de manera paralela, acelerando el entrenamiento, es por ello que la instalación de *CUDA* es necesaria. Con las tarjetas gráficas en funcionamiento se ha terminado de instalar el resto de las herramientas.

8.2. Configuración de la red *Restormer*

Con la plataforma SW correctamente desplegada, como se explica en la sección 8.1, se ha procedido a las pruebas de validación de la red. El primer paso ha sido descargarse el código de esta red de un repositorio de GitHub e instalarla en la plataforma SW generada. Tras ello, se comprueba que tanto la red como los códigos de entrenamiento funcionen correctamente. Para ello se han seguido los pasos especificados por los desarrolladores de

la red, los cuales se encuentran en los ficheros *readme.txt*. Debido a problemas con las versiones ha sido necesario la actualización de ciertas partes de los códigos. Además, se han modificado varios puntos del código para poder lanzar esta red de manera simultánea en las 8 tarjetas gráficas del servidor, permitiendo acelerar así el entrenamiento.

Tras estas pruebas se ha procedido a la modificación del código de la red para adaptarlo al problema presentado en el proyecto. El primer paso ha sido el cambio de la función de pérdidas a evaluar, se ha pasado de la evaluación de la métrica error absoluto al error (MAE) cuadrático medio (MSE). El cambio de función de pérdidas se realiza, ya que, en los algoritmos de estimación y cancelación de canal, es esta métrica *MSE* la que se evalúa. Por ello, es beneficioso que la función que se va a intentar minimizar en la red sea la misma. Tras esto se han cambiado los procesamientos de los datos de entrada haciéndolos mucho más simples, además se ha eliminado partes del código debido a que eran empleadas para la generación de los datos etiquetados. En el caso de uso esto no es necesario, ya que ya se cuenta con los datos etiquetados (el canal de loopback). En cuanto al entrenamiento de la red, se han usado los siguientes parámetros:

Tabla 2: Parámetros de entrenamiento de la red

Parámetros de entrenamiento	
Datos de entrada	128 x 128
N.º de datos de entrada	624
N.º de GPUs	8
Canales de entrada	1
Optimizador	AdamW
Función de pérdidas	MSE
Epochs	1000
Batch size	64

Tras estas modificaciones se ha pasado a la generación de los dataset (conjunto de datos) con los que se entrenará la red.

8.3. Generación y estructuración de los dataset

Los dataset generados en *Matlab* tienen una estructura de 2048×20000 . Partiendo de estos datos, y como la red trabaja con imágenes, se decidió dividir estos datos en secciones más pequeñas. Las comunicaciones simuladas en este proyecto tienen la característica de emplear un intervalo de guarda (IG) de $\frac{1}{8}$. El intervalo de guarda es un mecanismo empleado en las telecomunicaciones para evitar las interferencias en las transmisiones. Se encuentra al inicio del símbolo OFDM, siendo un tiempo donde no se manda información útil, sino que se espera a que lleguen los ecos del canal para que estos no interfieran en la comunicación. En el caso de estudio de las 2048 subportadoras OFDM $\frac{1}{8}$ son IG, es decir, 256. En estas 256 subportadoras se encuentra la información de canal. Por ello los datos se han cortado para solo utilizar las primeras 256 subportadoras. De esta manera se ha generado una nueva matriz de datos de 256×20000 . Esta matriz se seccionará

temporalmente en imágenes de longitud 256, generando así una serie de imágenes de tamaño 256×256 .

Las redes neuronales no trabajan con números complejos aún, debido a que la matemática que emplean no es capaz de interpretar los números complejos. Es por ello que antes de introducir los datos del canal es necesario generar unos nuevos datos que no consten de números complejos. Para ello se ha decidido dividir cada uno de los valores de canal en su parte real y su parte imaginaria. Esta información se concatena posteriormente en una única matriz, colocando primero los valores reales y después los valores imaginarios. Debido a este proceso y para mantener el tamaño de imagen de 256×256 , se ha decidido dividir la información del eje x en secciones de 128, partiendo así de imágenes de 256×128 con valores complejos a imágenes de 256×256 con valores simples. Estos datos 256×256 píxeles

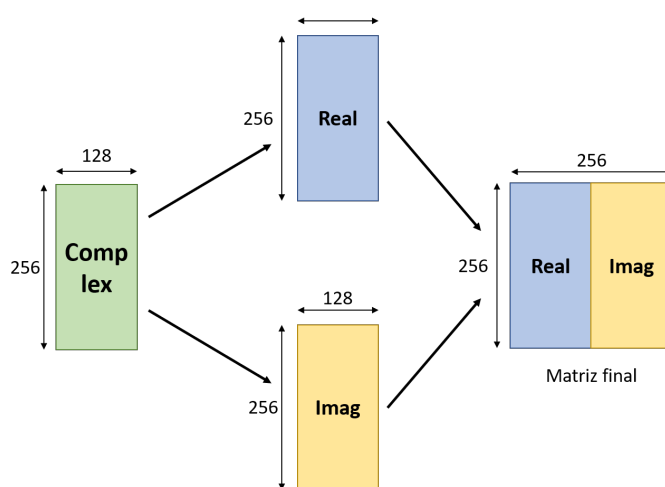


Figura 21: Generación de la matriz de entrada a la red

deberían ser la entrada de la red. Teniendo de este modo un dataset de 156 imágenes en total. Sin embargo, este tamaño de imagen de 256×256 hace que la red requiera de demasiada memoria y no sea posible llevar a cabo el entrenamiento por falta de esta en las GPUs del servidor. Por lo tanto, ha sido necesario reducir el tamaño de la matriz final de la *Figura 23* para poder entrenar a la red. Esta reducción se debe al aumento del coste computacional de las redes *transformers* con la longitud y anchura de las imágenes. Este coste aumenta cuadráticamente con ambas [31]. Por ello la matriz se ha dividido en 4 imágenes de 128×128 generando así un dataset de 624 imágenes. Puede parecer un inconveniente realizar esta división de la matriz, pero debido al funcionamiento de las ViT donde se mantienen las relaciones entre las imágenes no debería de afectar en el aprendizaje. Se han utilizado estos dataset de 624 imágenes como datos de entrenamiento para la red, a su vez se han utilizado otras 436 imágenes nuevas como datos de validación y otras 436 como imágenes de test.

Se han generado 3 tipos de datasets diferente para evaluar el comportamiento de la red en diferentes escenarios:

- **Escenario 1:** Canal directo estático y canal de loopback estático. Es el caso más simple debido a la invariabilidad de los datos en este escenario.

- **Escenario 2:** Canal directo con doppler de 5 Hz y canal de loopback estático. En este caso la variabilidad en el canal directo hace que toda la señal varíe como puede verse en la *Ecuación* (4).
- **Escenario 3:** Ambos canales cuentan con doppler, siendo este de 5 Hz en el directo y de 1 Hz en el de loopback. Es el caso más complejo, debido a la variabilidad existente en los datos.

8.4. Aproximación 1: PNG blanco/negro

La primera solución planteada fue seguir utilizando la red con sus herramientas de procesamiento de imagen. Es por ello que en este caso se decidió convertir las matrices de entrada a la red en imágenes con extensión PNG. Una de las características de las imágenes es el número de canales por los que se componen, existiendo 2 tipos de imágenes, imágenes a color, las cuales cuentan con 3 canales correspondientes a los colores RGB (rojo, verde y azul), e imágenes en blanco y negro, cuentan con un único canal. En esta situación, debido a que las matrices de los datos solo tienen una única capa, es decir, un único canal, se van a convertir estas matrices en imágenes PNG en blanco y negro.

Para convertir las matrices en imágenes en blanco y negro se ha realizado un script en MATLAB. Debido a que el formato de los PNG en blanco y negro son píxeles del tipo *unsigned int* de 8 bits, los valores de esta matriz deberán encontrarse entre 0 y 255. Para ello primero se han normalizado los datos de la siguiente forma:

$$x_n = \frac{x - m}{M - m} \quad (19)$$

Siendo m el valor mínimo de la señal x y M el máximo. Tras este paso se han multiplicado los datos por 255 y se han convertido al tipo *uint8*. Con esto se han generado las imágenes PNG en blanco y negro *Figura 22*. Tras el entrenamiento con estos datos se ha evaluado si la red ha funcionado correctamente, para ello se ha observado la estimación obtenida por la red comparándola con el canal limpia.

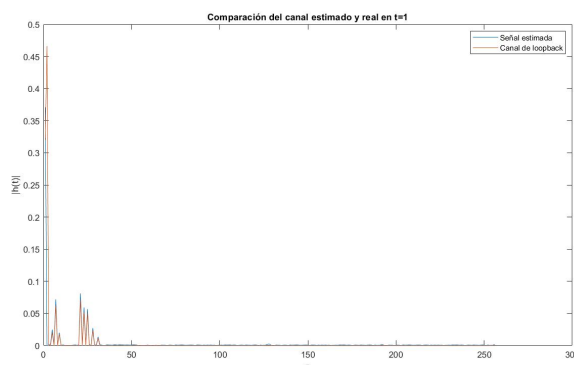
Este método solo se ha empleado con uno de los escenarios explicados en la sección 8.3, ya que al ser la primera solución pensada, esta se encuentra más orientada a demostrar que la red *Restormer* es capaz de usarse para este tipo de tareas. Además, no es una técnica correcta debido al uso de imágenes en blanco y negro, las cuales tienen una profundidad de 8 bits. El error introducido por cuantificación en este caso será $\frac{1}{2^8}$, considerando los valores en los que se mueven los datos del canal, este error no puede considerarse despreciable sino todo lo contrario. Por estas razones se decidió continuar hacia otras soluciones, como las presentadas en las secciones posteriores.



(a) Dato de entrada a la red en PNG

(b) Canal de loopback convertido a PNG

Figura 22: Datos de entrada a la red, (a) dato de entrada y (b) etiqueta del dato

Figura 23: Señal estimada y canal de loopback para datos de entrada PNG *grayscale*

8.5. Aproximación 2: PNG a color

La segunda solución fue convertir las matrices en imágenes a color, es decir, imágenes compuestas por 3 capas de color RGB. La decisión de uso de este nuevo tipo de dato se debe al aumento de profundidad de bit que este otorga. Pasando de la cantidad de $\frac{1}{2^8}$ a $\frac{1}{2^{24}}$, este aumento se debe al uso de 3 canales de color en lugar de uno. Cada uno de estos canales de color funciona del mismo modo que el canal explicado en la sección 8.4, pudiendo contener valores de 0 a 255, guardados en datos del tipo *wint8*. Por ello, el número de resolución de bit aumenta a 2^{24} al contar con 3 capas. Con objeto de convertir las matrices en imágenes PNG a color, primero se realizó un estudio de como PNG estructura estas imágenes. Como ya se ha mencionado, se usan 3 capas, cada una representando a uno de los colores RGB. La combinación de estas capas genera el color del pixel. La generación de estas imágenes es similar a la presentada en el punto anterior. En primer lugar, se han normalizado los datos con la *Ecuación (19)*, tras ello se han multiplicado los datos por 2^{24} generando así datos en ese umbral de valores. En las imágenes PNG los datos de la

capa R (Roja) son los 8 bits menos significativos, los de la capa G (Verde) los 8 siguientes y los 8 más significativos corresponden a la capa B (Azul). Se ha convertido el valor de decimal a binario y se han agrupado este en 8 bits. Esto seguirán el mapeo explicado, volviendo a convertirlos a decimal cuando se hayan dividido en las capas correspondientes. La combinación de estas 3 capas generará la imagen a color.

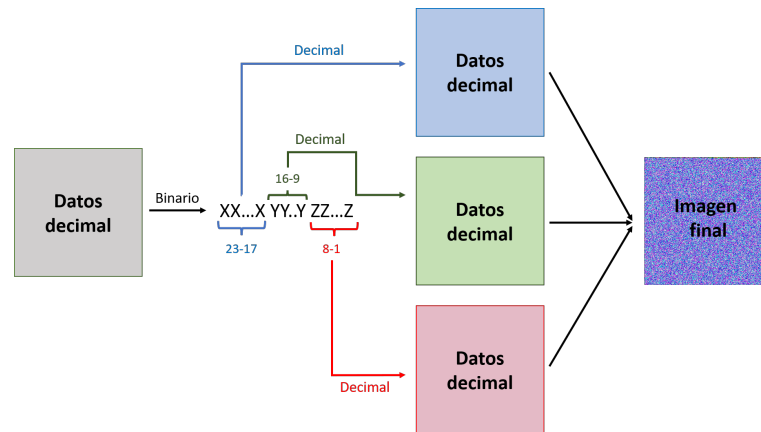


Figura 24: Proceso de creación de las imágenes a color

Con estas imágenes generadas se ha pasado al entrenamiento de la red, debido al aumento del número de capas en la imagen el coste computacional ha aumentado y ha sido necesario reducir el tamaño de las imágenes de entrenamiento. Se ha pasado de las imágenes 128 a imágenes 64×64 , por ello los datasets han aumentado su tamaño 4 veces. Este seccionamiento de las imágenes no supone un problema para el entrenamiento.

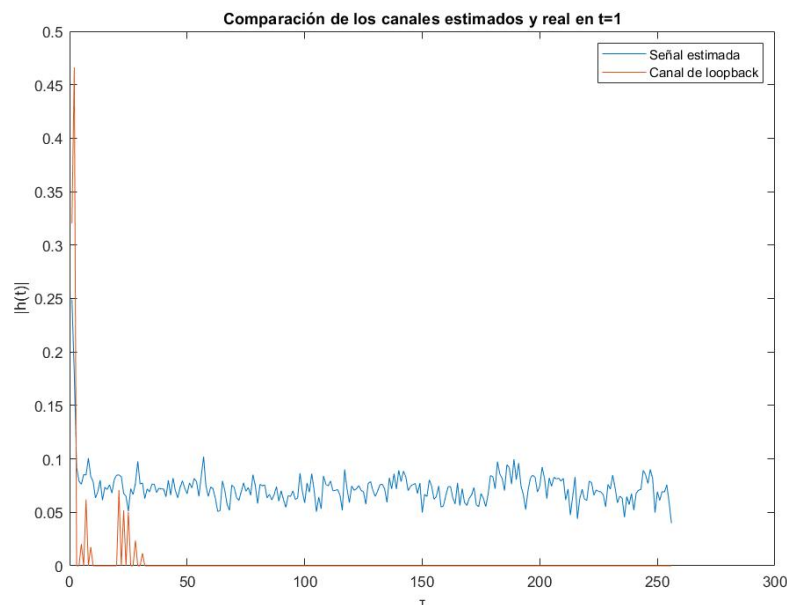


Figura 25: Comparación entre la estimación de la red y canal de loopback

Esta solución no es válida, como se aprecia en la *Figura 25* la red no estima correctamente ninguno de los ecos del canal, donde además se observa un ruido en las muestras en las que

no hay señal. Para comprobar de manera cuantitativa la validez de esta solución también se presenta el MSE obtenido tras el paso de los datos por la red.

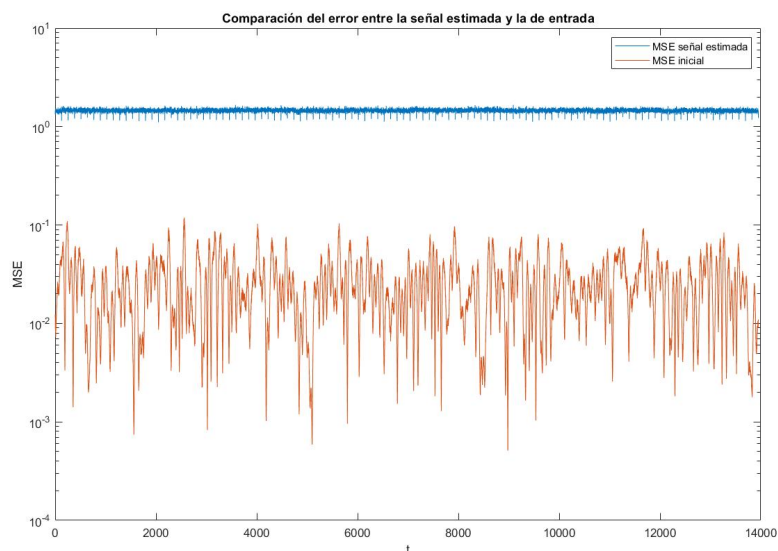


Figura 26: MSE de la señal estimada comparada con la señal original

En la *Figura 26* se observa como no solo la estimación no es buena, sino que empeora la situación inicial del sistema, por lo que esta solución fue descartada. Sin embargo, sí ha sido útil analizar los resultados obtenidos. En primer lugar, se ha descubierto que la red no realiza relación entre las distintas capas de colores. Esto sería de ayuda a la hora de la estimación debido a que los picos (ecos del canal) se detectarían mejor, ya que la información de estos se encuentra concentrada en las capas de mayor valor. Si relacionase estas capas, la red sería capaz de detectar los picos con mayor facilidad, pero como se observa en la *Figura 25* esto no ocurre. También se ha descubierto qué gran parte del error se debe a la codificación que se hace de la imagen. Al pasar de una imagen de tres capas a una matriz de 1 capa, los valores de estas se combinan. Debido a esto, un error en una de las capas produce una gran variación en el dato resultante de la combinación, magnificando de esta manera cualquier error que pudiese introducir la red.

Esta solución ha sido descartada y se ha buscado una donde ya no intervengan las imágenes, sino que se haga uso de los datos originales.

8.6. Aproximación 3: Matrices como dato de entrada

El uso de imágenes como datos de entrada no ha sido posible, es por ello que se ha modificado el código de la red para eliminar este procesamiento de imagen. Se ha editado el código de la *Restormer* para emplear como dato de entrada matrices, para ello se han modificado todas las funciones del código que usaban imágenes como datos. Las matrices que se van a emplear como dato son las generadas en la sección 8.3. Estas matrices cuentan con datos del tipo *float32*, es decir, contaremos con una precisión de $\frac{1}{2^{32}}$. En esta aproximación se han evaluado los 3 escenarios mencionados, siendo el primer escenario el caso de ningún canal con doppler.

8.6.1. Escenario 1: Canales sin doppler

Este escenario es el más simple debido al comportamiento estático de ambos canales. En este caso se han pasado los datos de entrada por la red modificada obteniendo la siguiente salida.

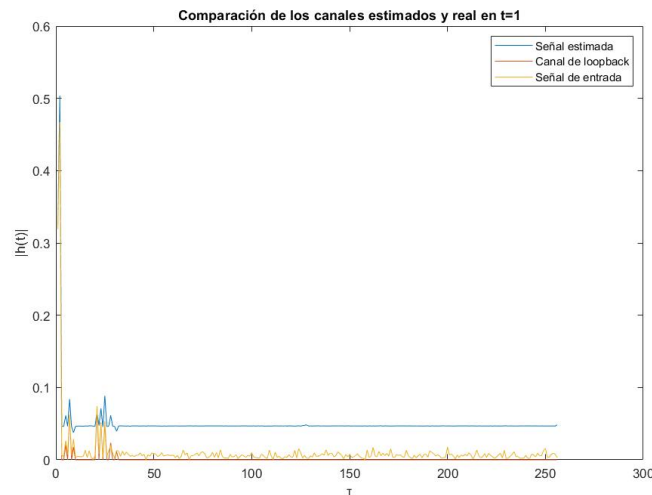
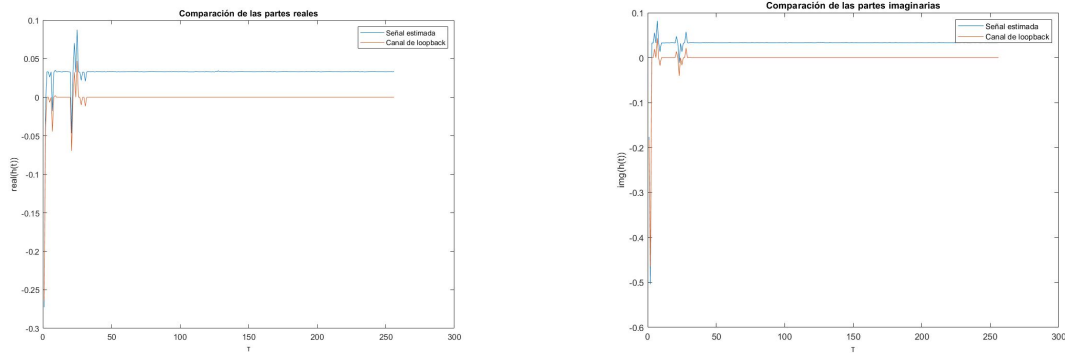


Figura 27: Señal estimada, canal de loopback y señal de entrada a la red

Esta señal estimada no se ajusta al canal de loopback por lo que parece que la red estima erróneamente el canal. Sin embargo, se observa que la señal se encuentra desplazada un valor constante de la deseada. Esto puede apreciarse observando a partir de $\tau = 50$ donde la señal es prácticamente plana, pero desplazado un valor del 0, que es el valor que debería tener. Este valor también se aprecia en los ecos detectados donde también se encuentran desplazados. Debido a ello se ha realizado un estudio de los datos más exhaustivo. Se ha partido de los datos de salida de la red, los cuales son la parte real e imaginaria del canal por separado, como se observa en la *Figura 23*. En estos datos se ha observado que la forma de la señal se realiza de manera muy similar, pero esta se encuentra desplazada un valor cuasi constante.



(a) Parte real

(b) Parte imaginaria

Figura 28: Comparación entre las partes reales e imaginarias de las señales

Se ha realizado un estudio de los datos donde se ha observado que el valor de desplazamiento es prácticamente constante en todos los instantes temporales del canal. Debido a su determinismo, este valor puede ser restado y obtener una estimación que se ajuste verdaderamente al canal. Para obtener el valor que minimiza el error en la estimación se ha realizado un análisis estadístico, el valor que disminuye el error es la moda de toda la señal. Con este método se han calculado dos valores, uno para la parte real y otro para la imaginaria. Estos valores han sido restados a la señal generando de esta manera la nueva estimación, la cual se adecua mucho más al canal de loopback deseado. En las siguientes figuras solo se muestran las 50 primeras subportadoras, para observar la estimación con mayor detalle, debido a que el resto de la señal es nulo.

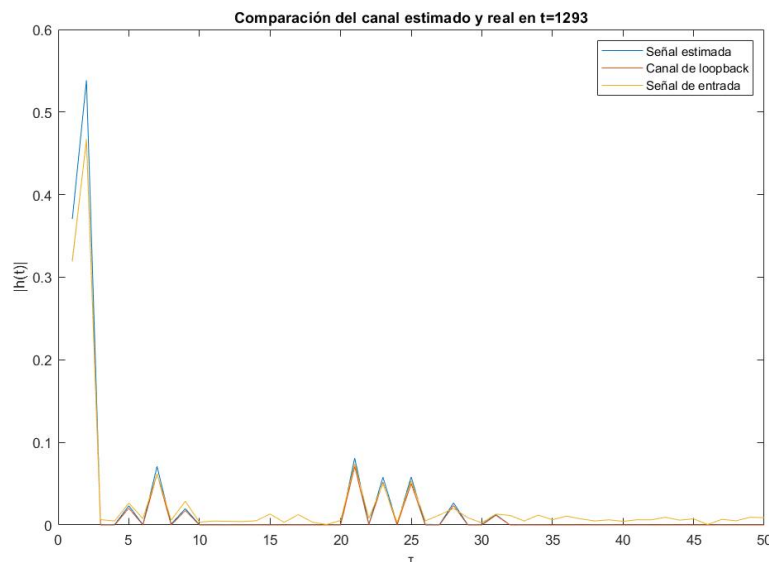


Figura 29: Señal estimada, canal de loopback y señal de entrada a la red tras la corrección

La señal estimada se ajusta bastante bien al canal de loopback detectando correctamente todos los ecos de este. Además, elimina el resto del ruido de la señal. También se observa

que el primer eco, el eco principal, es el punto que peor se estima de la señal. Lo correcto sería realizar un estudio acerca de la causa de este suceso, pero debido a la falta de tiempo se ha considerado suficientemente bueno el resultado y se ha pasado al caso de estudio de otros escenarios. También se ha estudiado la métrica del MSE para poder tener un análisis cuantitativo de la calidad de la estimación.

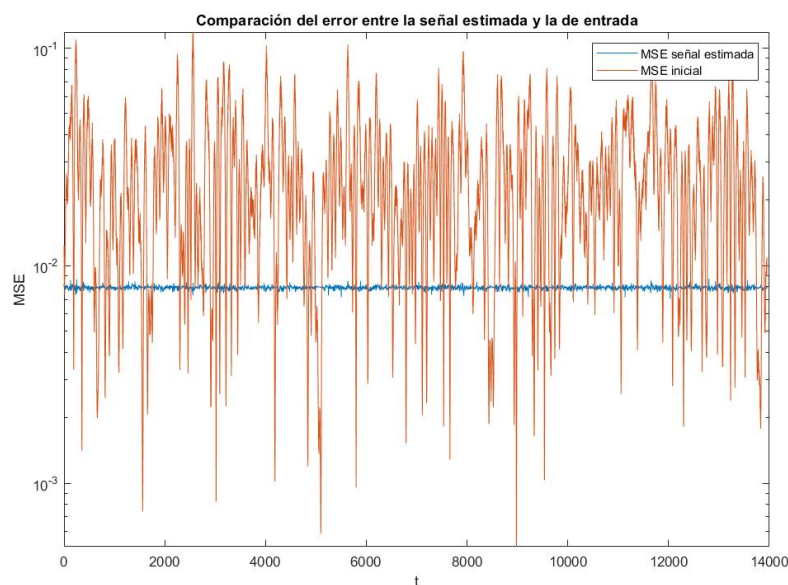


Figura 30: Análisis del MSE del escenario 1

El MSE de la señal estimada se encuentra por debajo del original en la mayor parte del tiempo (*Figura 30*). A pesar de ello, se observa que no existe una gran mejoría respecto a la señal inicial. La mayor parte del error existente en la señal estimada se encuentra en los primeros instantes, donde la estimación no se adecua correctamente a la forma del eco principal (ver *Figura 29*). La calidad de la estimación sigue siendo buena como se ha observado en la *Figura 29*, teniendo un buen ajuste de la señal estimada a los ecos del canal de loopback, es por ello que la estimación del canal se considera válida. El MSE de este escenario es engañoso al tratarse de una situación muy favorable debido al comportamiento estático de ambos canales. Esto hace que el MSE de partida de la señal tenga puntos donde es muy bajo.

8.6.2. Escenario 2: Canal directo con doppler

En este escenario se ha realizado el entrenamiento de la red para el caso de un canal directo con 5 Hz de desplazamiento doppler. Este desplazamiento doppler afecta a los ecos del canal directo h_{FW} , debido a ello los valores de la señal de entrada variarán en mayor medida que en el escenario 1. Esta variabilidad se justifica observando la *Ecuación* (4) donde el dato de entrada depende de dicho canal. El número de datos de entrenamiento ha sido el mismo que para el escenario 1, sin embargo, para los datos de validación se ha contado con un conjunto de datos mayor.

A la señal estimada por la red se le ha aplicado la misma corrección del escenario 1, debido a la presencia del valor de desplazamiento constante. A continuación se muestra la señal estimada final.

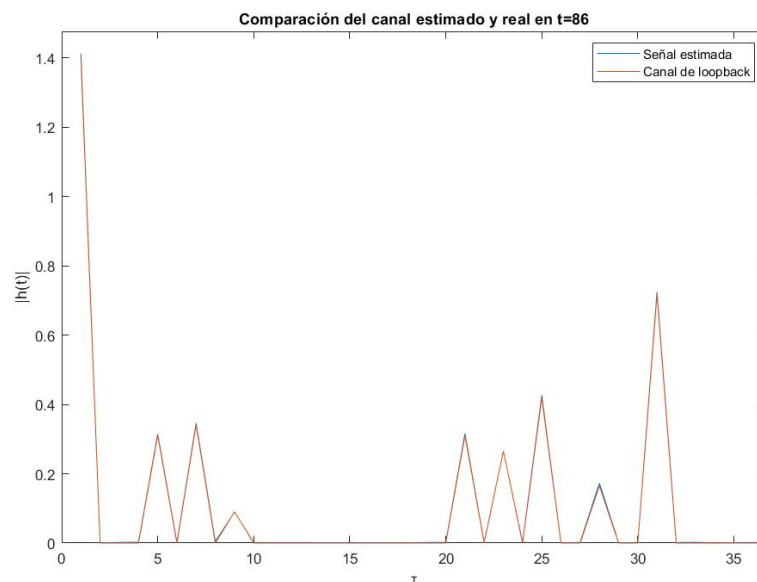


Figura 31: Señal estimada y canal de loopback (escenario 2)

En la *Figura 31* se observa como la señal se ajusta a la señal estimada de manera muy precisa, haciendo difícil la diferenciación entre las señales estimada y real. Para analizar cuantitativamente esta estimación se presenta a continuación la métrica del MSE.

En la *Figura 32* observamos la mejora que presenta el uso de la red en este escenario. El MSE se encuentra aproximadamente 2 órdenes de magnitud por debajo del MSE de la situación de partida. Se observa que el MSE de partida presenta un mayor valor, debido al doppler existente en la señal directa, el cual empeora la calidad de la señal recibida. Sin embargo, observamos como para casos más variables la red parece estimar de manera más precisa.

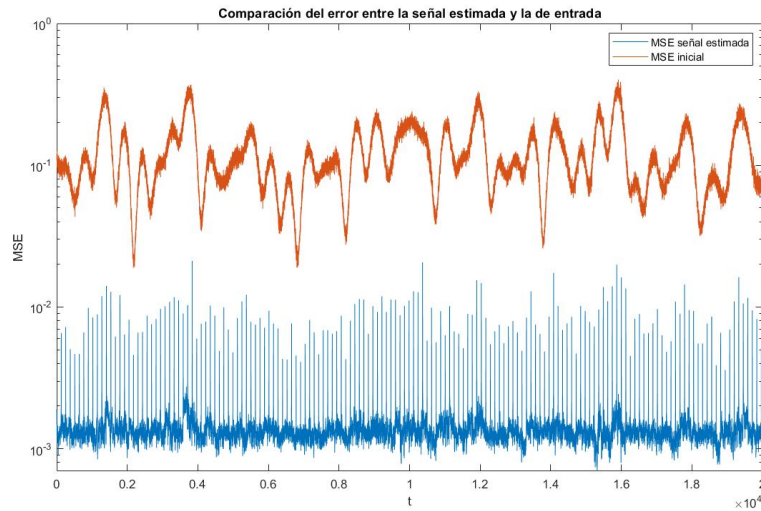


Figura 32: Análisis del MSE del escenario 2

8.6.3. Escenario 3: Canal directo y de Loopback con doppler

En este escenario el canal directo contará con un doppler de 5 Hz y el canal de loopback con 1 Hz. Con estos valores de partida se han generado los datos de entrada de la misma manera que en los escenarios 1 y 2, siguiendo además el mismo entrenamiento. Estos datos son los que mayor variabilidad presentan, debido a que dos componentes de la *Ecuación* (4) cuentan con variabilidad doppler. La señal estimada por la red ha recibido el mismo procesamiento que las señales de los escenarios 1 y 2. Se ha contado con el mismo número de datos para entrenamiento y validación que el escenario 2. La estimación de la señal en este escenario se presenta en la siguiente figura.

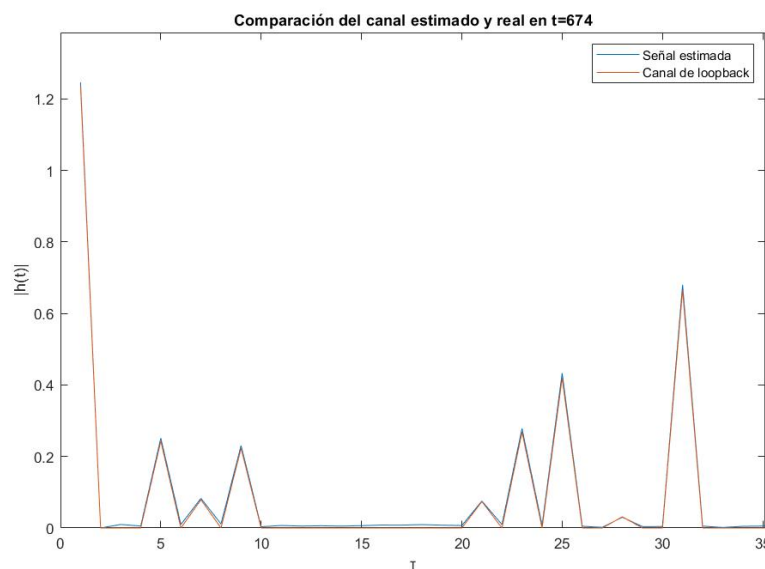


Figura 33: Señal estimada y canal de loopback (escenario 3)

La señal estimada se ajusta de manera bastante precisa al canal de loopback, sin embargo, se observa un mayor error que en el escenario 2. Para analizar cuantitativamente esta estimación se presenta a continuación la métrica del MSE.

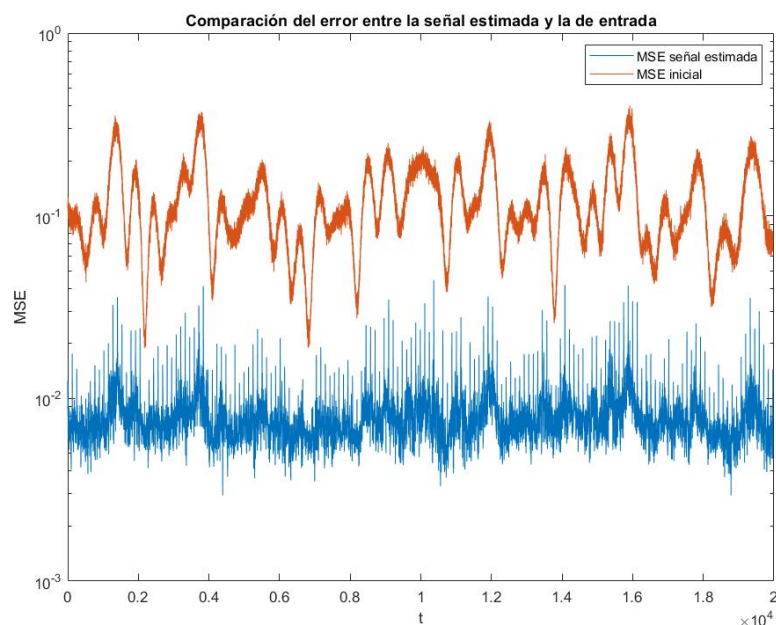


Figura 34: Análisis del MSE del escenario 3

En la *Figura 34* observamos la mejora que presenta el uso de la red en este escenario. El MSE de partida es el peor de los 3 escenarios. El error de la señal estimada se encuentra aproximadamente 1 orden de magnitud por debajo de este. La estimación del canal es buena al tratarse del escenario más adverso. Además, analizando los datos en mayor profundidad se ha observado que las zonas donde el canal debería nulo es donde se introduce el mayor error, esto no es un gran inconveniente debido a la posibilidad de introducir un parámetro que convierta en 0 estas zonas.

8.6.4. Mejoras a futuro

Realizando un análisis más en detalle de los resultados obtenidos se han observado ciertas características que han otorgado algunas ideas para la mejora de los resultados.

En primer lugar, el procesado realizado a la señal tras la salida de la red podría ser eliminado. Para ello se pueden introducir dos nuevos parámetros en la red que representen este desplazamiento de los datos reales e imaginarios y que la red se encargue de minimizarlos.

También se ha observado en los MSE de la señal que existe un pico de error cada 128 muestras de la señal. Esto se debe al tamaño de las imágenes donde al tratarse de secciones de 128×128 la red cuenta con menos información para realizar la estimación. Este pico de error puede observarse en la *Figura 35* donde se ha realizado una ampliación de la señal MSE.

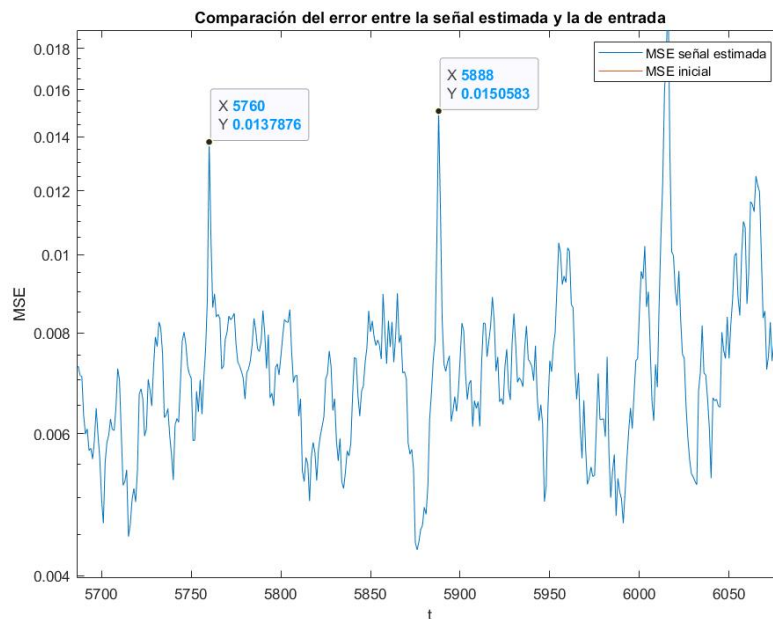


Figura 35: Zoom en la señal de MSE

Debido a esta característica se ha pensado emplear una nueva estructuración de los datos de entrada donde esto no ocurra. En ella se generarán nuevas matrices de datos de entrada, las cuales tendrán en el centro el instante temporal que se desea estimar. Es decir, estas nuevas imágenes contarán con 63 instantes anteriores y posteriores que no se estimarán. Se generarán 20000 matrices como datos de entrada de la red, aumentando así la dimensión del conjunto de datos.

9. Plan de proyecto

En este apartado se presenta las diferentes etapas planteadas y seguidas para la consecución de este proyecto. Se planteó con una duración aproximada de 7 meses, siendo este dividido en diferentes etapas, estas etapas han sido representadas mediante paquetes de trabajo. Del mismo modo, cada paquete de trabajo está compuesto por diferentes tareas. En este apartado se pretende mostrar las tareas seguidas, así como el tiempo que se ha invertido en cada una de ellas.

9.1. Paquetes de trabajo y tareas

Paquete de trabajo 1 → Coordinación, gestión y supervisión del proyecto

Tarea 1.1: Supervisión y seguimiento del proyecto.

Esta tarea representa el seguimiento a realizar sobre el trabajo a lo largo del desarrollo del proyecto. Va a ser realizada durante la duración completa de este. En esta tarea se incluyen las reuniones con el director del proyecto y los subdirectores, las cuales servirán para realizar un seguimiento del trabajo realizado, resolver las dudas que surjan durante las etapas y marcar el camino que se deberá seguir para poder terminar el proyecto en el plazo establecido.

Paquete de trabajo 2 → Familiarización con el problema

Tarea 2.1: Contextualización con el marco del proyecto

Conocimiento del contexto en el que se encuentra enmarcado el proyecto y de la problemática que se pretende resolver.

Tarea 2.2: Búsqueda bibliográfica

Recopilación de diferentes documentos y fuentes de información que sean de utilidad para el comienzo del desarrollo del proyecto.

Tarea 2.3: Introducción a las herramientas de desarrollo SW

Estudio de las herramientas SW a emplear durante el proyecto. Se realizarán tareas de replicación de códigos ya existente con las herramientas a estudiar.

Paquete de trabajo 3 → Análisis de soluciones ya existentes

Tarea 3.1: Análisis de técnicas de procesado de señal

En esta tarea se va a realizar un análisis de los métodos tradicionales empleados para la resolución del problema de autointerferencia.

Paquete de trabajo 4 → Análisis de soluciones basadas en IA

Tarea 4.1: Análisis de diferentes algoritmos de IA

Estudio de las diferentes arquitecturas y técnicas de IA que pueden ser empleadas para la realización del proyecto. Se analizan diferentes familias de IA, para seleccionar finalmente una.

Tarea 4.2: Análisis de la familia de IA seleccionada

Análisis de las diferentes arquitecturas presentes en la familia seleccionada. Basado en las ventajas y desventajas que presenta cada uno para el problema planteado.

Tarea 4.3: Selección de la arquitectura final

Dentro del análisis de la tarea 4.2 se seleccionará la arquitectura que se considera más óptima y se resolverá el problema con ella.

Paquete de trabajo 5 → Desarrollo de la plataforma SW**Tarea 5.1: Instalación de las herramientas SW**

Instalación las herramientas SW necesarias para la realización del proyecto dentro del servidor de cálculo. Se instalarán las versiones necesarias y se comprobará la compatibilidad y correcto funcionamiento de estas herramientas.

Tarea 5.2: Configuración y puesta en marcha de la red

Instalación de la red *Restormer* y de las dependencias de esta. Se instalarán los controladores de las tarjetas gráficas y se comprobará su funcionamiento.

Hito 1: Implementación completa de la red *Restormer***Paquete de trabajo 6 → Pruebas y validación****Tarea 6.1: Familiarización con la red**

Se utilizará la red para replicar tareas ya resueltas. De este modo se comprenderá el funcionamiento de la red y del código, así como se comprobará el correcto funcionamiento de la misma

Tarea 6.2: Obtención de resultados

Se obtendrá una estimación de canal usando la red *Restormer*.

Tarea 6.3: Análisis de resultados

Se extraerán las conclusiones pertinentes a partir de los resultados obtenidos. Para ello, se evaluarán unas métricas seleccionadas. Estas conclusiones servirán para evaluar la calidad de la solución y determinar posibles mejoras futuras.

Paquete de trabajo 7 → Redacción del documento**7.1 → Redacción del documento**

Redacción del documento final en el que se presentará el proyecto realizado.

Hito 2 → Entrega del documento

9.2. Diagrama de Gantt

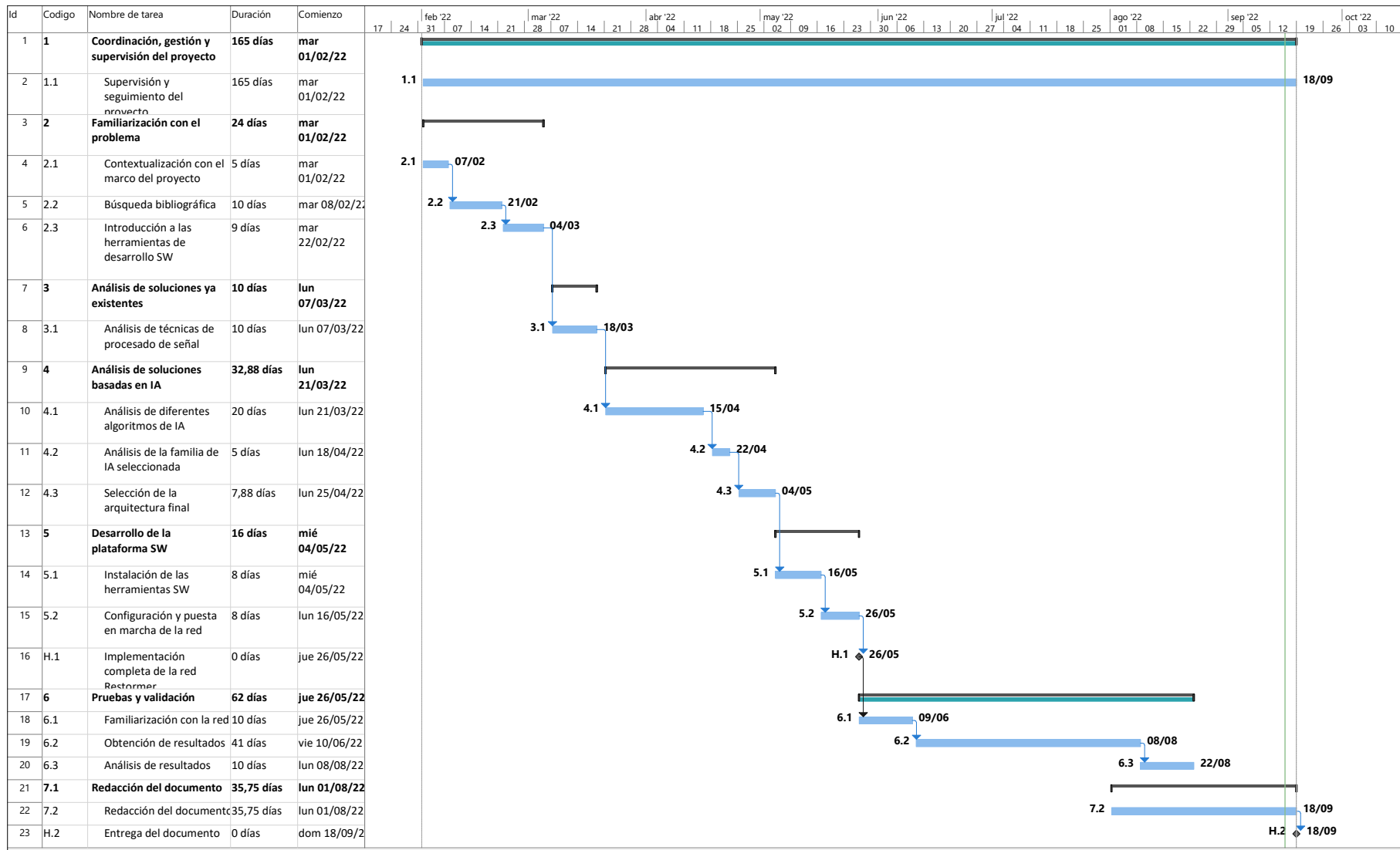


Figura 36: Diagrama de Gantt

10. Conclusiones

Tras el estudio y análisis de la bibliografía relacionada con las técnicas de IA, se ha llegado a la conclusión de que aún quedan numerosas técnicas por probar para la resolución del problema. Además, el hecho de la mejora y simplicidad que aportarían estas técnicas al esquema de recepción de este tipo de torres, hace muy interesante su estudio. Por otro lado, en cuanto a la técnica seleccionada, se puede determinar que funciona para este problema de cancelación de la autointerferencia. No se ha podido desarrollar el estudio de esta técnica todo lo deseado, pero si se han podido extraer varias conclusiones de la realización del proyecto.

- Se ha realizado una buena investigación donde se han identificado una gran cantidad de posibles soluciones basadas en IA, basadas en las características presentes en este problema de autointerferencia.
- La solución seleccionada es apta para la estimación del canal de loopback, mostrando un mejor rendimiento en los escenarios de comunicación más adversos. Sin embargo, teniendo en cuenta el número de parámetros de la red *Restormer* y su complejidad, el resultado obtenido no es tan bueno como el esperado.
- La falta de tiempo ha propiciado que no haya sido posible el análisis de la solución elegido hasta el final, esto hace que las conclusiones obtenidas sobre la validez de la solución no puedan considerarse absolutas. Dejando abierto un camino de investigación el cual sería interesante recorrer para poder generar unas conclusiones finales y absolutas.
- Debido a los beneficios que puede aportar la IA y en vista de las numerosas técnicas existentes, se podrán abrir nuevas líneas de investigación que analicen la viabilidad de todas las técnicas no exploradas aún.

11. Bibliografía

- [1] L. Zhang, W. Li, Y. Wu et al., “Using Layered Division Multiplexing for Wireless In-Band Distribution Links in Next Generation Broadcast Systems,” *IEEE Transactions on Broadcasting*, vol. 67, n.º 1, págs. 68-82, 2021. DOI: 10.1109/TBC.2020.2989638.
- [2] Z. H. Hong, L. Zhang, W. Li et al., “Blind RF Self-Interference Cancellation for In-Band Distribution Link in ATSC 3.0,” en *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2022, págs. 1-6. DOI: 10.1109/BMSB55706.2022.9828730.
- [3] W. Li, L. Zhang, Y. Wu et al., “Integrated Inter-Tower Wireless Communications Network for Terrestrial Broadcasting and Multicasting Systems,” *IEEE Transactions on Broadcasting*, vol. 67, n.º 3, págs. 570-581, 2021. DOI: 10.1109/TBC.2021.3081861.
- [4] L. Zhang, W. Li, Y. Wu et al., “Layered-division-multiplexing: Theory and practice,” *IEEE Transactions on Broadcasting*, vol. 62, n.º 1, págs. 216-232, 2016.
- [5] E. Iradier, I. Bilbao, J. Montalban et al., “Analog Cancellation in ATSC 3.0 for Enabling Inter-Tower Communications Network,” en *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2021, págs. 1-6. DOI: 10.1109/BMSB53066.2021.9547177.
- [6] S. Bojja Venkatakrishnan, E. A. Alwan y J. L. Volakis, “Wideband RF Self-Interference Cancellation Circuit for Phased Array Simultaneous Transmit and Receive Systems,” *IEEE Access*, vol. 6, págs. 3425-3432, 2018. DOI: 10.1109/ACCESS.2017.2788179.
- [7] L. Anttila, D. Korpi, V. Syrjälä y M. Valkama, “Cancellation of power amplifier induced nonlinear self-interference in full-duplex transceivers,” en *2013 Asilomar Conference on Signals, Systems and Computers*, 2013, págs. 1193-1198. DOI: 10.1109/ACSSC.2013.6810482.
- [8] E. Ahmed y A. M. Eltawil, “On Phase Noise Suppression in Full-Duplex Systems,” *IEEE Transactions on Wireless Communications*, vol. 14, n.º 3, págs. 1237-1251, 2015. DOI: 10.1109/TWC.2014.2365536.
- [9] A. Canziani e Y. LeCun, *NYU Deep Learning, Spring 2020*, <https://atcold.github.io/pytorch-Deep-Learning/>, [Online; accessed], 2020.
- [10] H. Ye, G. Y. Li y B.-H. Juang, “Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems,” *IEEE Wireless Communications Letters*, vol. 7, n.º 1, págs. 114-117, 2018. DOI: 10.1109/LWC.2017.2757490.
- [11] K. Burse, R. N. Yadav y S. C. Shrivastava, “Channel Equalization Using Neural Networks: A Review,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, n.º 3, págs. 352-357, 2010. DOI: 10.1109/TSMCC.2009.2038279.
- [12] T. Wang, C.-K. Wen, S. Jin y G. Y. Li, “Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels,” *IEEE Wireless Communications Letters*, vol. 8, n.º 2, págs. 416-419, 2019. DOI: 10.1109/LWC.2018.2874264.

- [13] C.-K. Wen, W.-T. Shih y S. Jin, "Deep Learning for Massive MIMO CSI Feedback," *IEEE Wireless Communications Letters*, vol. 7, n.º 5, págs. 748-751, 2018. DOI: 10.1109/LWC.2018.2818160.
- [14] K. Salehian, M. Guillet, B. Caron y A. Kennedy, "On-channel repeater for digital television broadcasting service," *IEEE Transactions on Broadcasting*, vol. 48, n.º 2, págs. 97-102, 2002. DOI: 10.1109/TBC.2002.1021274.
- [15] S. I. Park, S. R. Park, H. Eum, J.-y. Lee, Y.-T. Lee y H. M. Kim, "Equalization On-Channel Repeater for Terrestrial Digital Multimedia Broadcasting System," *IEEE Transactions on Broadcasting*, vol. 54, n.º 4, págs. 752-760, 2008. DOI: 10.1109/TBC.2008.2001711.
- [16] M. M. Wang, "Dynamic Gain Management for On-Channel Repeaters," *IEEE Transactions on Broadcasting*, vol. 59, n.º 4, págs. 685-692, 2013. DOI: 10.1109/TBC.2013.2284417.
- [17] F. Zabini, M. Mazzotti, D. Dardari, G. Chiurco y O. Andrisano, "Performance and Stability Analysis of Echo Cancellers Based on Training Sequences," *IEEE Transactions on Broadcasting*, vol. 60, n.º 3, págs. 437-451, 2014. DOI: 10.1109/TBC.2014.2326052.
- [18] K. O'Shea y R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.
- [19] X.-J. Mao, C. Shen e Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," *arXiv preprint arXiv:1606.08921*, 2016.
- [20] C. Dong, C. C. Loy, K. He y X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, n.º 2, págs. 295-307, 2015.
- [21] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, n.º 3, págs. 211-252, 2015.
- [22] K. Simonyan y A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell y S. Xie, "A convnet for the 2020s," en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, págs. 11 976-11 986.
- [24] V. Feng, "An overview of resnet and its variants," *Towards data science*, vol. 2, 2017.
- [25] B. Lindemann, T. Muller, H. Vietz, N. Jazdi y M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, págs. 650-655, 2021, 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2021.03.088>. direcci3n: <https://www.sciencedirect.com/science/article/pii/S2212827121003796>.
- [26] T. Mikolov, S. Kombrink, L. Burget, J. Černocky y S. Khudanpur, "Extensions of recurrent neural network language model," en *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, págs. 5528-5531. DOI: 10.1109/ICASSP.2011.5947611.

- [27] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [28] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan y M. Shah, “Transformers in vision: A survey,” *ACM Computing Surveys (CSUR)*, 2021.
- [29] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu y H. Li, “Uformer: A general u-shaped transformer for image restoration,” en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, págs. 17 683-17 693.
- [30] M. Chang, Q. Li, H. Feng y Z. Xu, “Spatial-Adaptive Network for Single Image Denoising,” *arXiv preprint arXiv:2001.10291*, 2020.
- [31] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan y M.-H. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, págs. 5728-5739.