



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO INGENIERÍA MECÁNICA



**Diagnóstico de fallas en rodamientos mediante imágenes tiempo-frecuencia
utilizando aprendizaje profundo**

POR

Cristian Alejandro Bastias Ramirez

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para
optar al título profesional de Ingeniero Civil Mecánico.

Profesor Guía:
Dr.-Ing. Félix Leaman W.

Marzo 2023
Concepción (Chile)

© 2023 Cristian Alejandro Bastias Ramirez

© 2023 Cristian Alejandro Bastias Ramirez

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o
procedimiento, incluyendo la cita bibliográfica del documento

Agradecimientos

Agradezco en primer lugar a mi familia y su apoyo incondicional durante mi proceso de formación, entregándome las herramientas para desarrollarme sin dificultades. También, agradezco a todas las personas y amigos cercanos que encontré durante este proceso, así también a mis mascotas que me ayudaban haciendo compañía durante las largas horas de estudio.

También, agradecer la ayuda de mi profesor guía don Félix Leaman, por ayudarme y entregarme la orientación necesaria durante todo el proceso de desarrollo de esta tesis.

Resumen

Los rodamientos son cruciales para el funcionamiento de las máquinas rotatorias, por lo que en esta memoria se desarrolla una metodología que utiliza las *Convolutional Neural Networks* (CNN) para clasificar de manera automática imágenes tiempo-frecuencia, que son generadas por la *Short-Time Fourier Transform* (STFT) y se encuentran asociadas a distintos modos de falla del rodamiento. Esta metodología se desarrolla con tal de entregar una herramienta que permita mejorar la confiabilidad de estos componentes, mediante la aplicación de inteligencia artificial.

La aplicación de esta metodología contempla la evaluación de la influencia asociada al uso de distintas configuraciones, tales como la duración de la señal, normalización de colores y ancho de ventana de la STFT. La duración de la señal afecta directamente en la cantidad de imágenes disponibles para el entrenamiento. Donde una mayor cantidad de imágenes mejora el desempeño de la CNN, mientras que una baja disponibilidad afecta negativamente al desempeño de esta. En cuanto a la normalización de colores, se evalúan tres tipos de normalización: local, global y por señal. La normalización local obtiene los mejores resultados, puesto que resalta de manera efectiva las características representativas asociadas a los modos de falla del rodamiento. Posteriormente, para la configuración final se tiene que un ancho de ventana pequeño presenta los mejores los resultados. Siendo una de las configuraciones más importantes con tal de aumentar la robustez y generalización del modelo resultante. Entre otras consideraciones, la selección correcta de hiperparámetros de la CNN contribuye a mitigar la influencia de la inicialización aleatoria de pesos, y por ende ayuda a disminuir la variabilidad de los resultados de la CNN. Finalmente, las mejores configuraciones corresponden a una duración de 10 giros del eje, normalización local y ancho de ventana de 4096 puntos de la señal, para las que se genera un modelo que alcanza un 100% de precisión en la clasificación de estas imágenes tiempo-frecuencia.

Entre otros aportes de esta memoria, se encuentra la evaluación de factibilidad del uso de *Transfer Learning* (TL) entre distintos bancos de datos de rodamientos. Lo que se realiza con el fin abordar la problemática asociada a la escasez de datos de falla disponibles, una situación común en la industria. Estos bancos de datos constan de distintas condiciones de operación y modos de falla en común. El TL se realiza utilizando el conocimiento previamente adquirido durante el entrenamiento de la CNN en el banco de datos de la Universidad de Ottawa, considerándose las mejores configuraciones mencionadas. Posteriormente, este conocimiento se trasfiere a un nuevo banco de datos de la *American Society for Machinery Fault Prevention Technology* mediante técnicas de TL, como el *Fine-Tuning*. Para lo que se obtiene como resultado una mayor velocidad de entrenamiento y convergencia del modelo, con una precisión promedio del $100\% \pm 0,00$ y presentándose como solución ante una baja cantidad de datos disponibles de falla. Lo que convierte el TL como una alternativa factible para su aplicación en la industria.

Palabras claves: Inteligencia Artificial, Redes Neuronales Convolucionales, Rodamientos, Transferencia de conocimiento, Transformada Corta de Fourier, Mantenimiento predictivo.

Abstract

Bearings are crucial for the operation of rotating machines, so in this thesis is developed a methodology that uses Convolutional Neural Networks (CNN) to automatically classify time-frequency images, which are generated by the Short-Time Fourier Transform (STFT) and are associated with different bearing failure modes. This methodology is developed in order to deliver a tool that allows improving the reliability of these components, through the application of artificial intelligence.

The application of this methodology contemplates the evaluation of the influence associated with the use of different configurations, such as the duration of the signal, normalization of colors and window width of the STFT. The signal duration directly affects the number of images available for training. Where a greater number of images improves the performance of CNN, while low availability negatively affects its performance. In respect to color normalization, three types of normalization are evaluated: local, global and by signal. Local normalization achieves the best results, since it effectively highlights representative characteristics associated with bearing failure modes. Subsequently, for the final configuration, a small window width presents the best results. Being one of the most important configurations in order to increase the robustness and generalization of the resulting model. Among other considerations, the correct selection of CNN hyperparameters contributes to mitigate the influence of random initialization of weights, and therefore contributes to reduce the variability of the CNN results. Finally, the best configurations correspond to a duration of 10 revolutions of the shaft, local normalization and window width of 4096 signal points, for which a model is generated that reaches 100% accuracy in the classification of these time-frequency images.

Among other contributions of this memory, there is the evaluation of the feasibility of the use of TL between different datasets of bearings. Which is done in order to address the problems associated with the scarcity of available failure data, a common situation in the industry. These datasets consist of different operating conditions and failure modes in common. The TL is carried out using the knowledge previously acquired during the training of the CNN in the dataset of the University of Ottawa, considering the best configurations mentioned. Subsequently, this knowledge is transferred to a new dataset of the American Society for Machinery Fault Prevention Technology using TL techniques, such as Fine-Tuning. For which, as a result, a greater training speed and convergence of the model is obtained, with an average precision of $100\% \pm 0.00$ and presenting itself as a solution to a low amount of available failure data. Which turns the use of TL as a feasible alternative for its application in the industry.

Keywords: Artificial Intelligence, Convolutional Neural Networks, Bearings, Transfer Learning, Short-time Fourier Transform, Predictive maintenance.

Tabla de Contenidos

Agradecimientos.....	i
Resumen	ii
Abstract	iii
Tabla de Contenidos.....	iv
Lista de Tablas	vi
Lista de Figuras	vii
Acrónimos	x
CAPÍTULO 1: Introducción.....	1
1.1 Planteamiento del problema	1
1.2 Estado del arte	2
1.3 Hipótesis	4
1.4 Metodología.....	4
1.5 Objetivo general	5
1.6 Objetivos específicos.....	5
CAPÍTULO 2: Marco teórico.....	6
2.1 Redes Neuronales Convolucionales.	6
2.1.1 Capa convolucional.....	6
2.1.2 Función de activación.	7
2.1.3 Capa de Pooling (Pooling Layer).....	8
2.1.4 Capa completamente conectada y clasificador Softmax.....	9
2.1.5 Gradiente descendente y función de costo.....	9
2.2 Transformadas Tiempo – Frecuencia.	10
2.3 Transformada Corta de Fourier	11
2.3.1 Función de ventana.	12
2.3.2 Ancho de ventana.....	13
2.4 Normalización	15
2.4.1 Normalización global.....	15
2.4.2 Normalización local	16
2.4.3 Normalización por señal	16
CAPÍTULO 3: Casos de estudio y metodología de trabajo.	19

3.1 Banco de simulación de fallas.	19
3.1.1 Banco de datos.	20
3.2 Metodología de trabajo.	21
3.2.1 Generación de bancos de imágenes.	21
3.2.2 Arquitectura y entrenamiento de la CNN.	22
3.2.3 Selección de la mejor configuración para generar bancos de imágenes mediante STFT. .	25
CAPÍTULO 4: Resultados y discusión.	27
4.1 Duración de señal.	27
4.1.1 Resultados de precisión.	27
4.1.2 Resultados de precisión y pérdida por épocas.	29
4.1.3 Resultados de test para mejores modelos.	31
4.1.4 Evaluación influencia hiperparámetros en entrenamiento de la CNN.	32
4.2 Resultados para distintos anchos de ventana y normalizaciones.	36
4.3 Reevaluación de la duración de señal y anchos de ventanas.	39
4.4 Visualización de características aprendidas mediante t-SNE.	41
4.5 Visualización de mapas de características para distintos tipos de falla.	43
4.6 Comparación con modelo de regresión logística.	46
CAPÍTULO 5: <i>Transfer Learning</i>	47
5.1 Contexto.	47
5.2 Banco de datos objetivo y metodología.	48
5.3 Generación de banco de imágenes.	49
5.3.1 Metodología para la aplicación de TL.	49
5.4 Resultados y discusión.	50
5.4.1 Evaluación de la influencia en la cantidad de datos disponibles aplicando TL.	50
5.4.2 Comparativa de transferencia de conocimiento y entrenamiento desde cero.	53
5.4.3 Limitaciones actuales para la aplicación de transferencia de conocimiento.	55
CAPÍTULO 6: Conclusiones y recomendaciones.	56
Referencias.	57
Anexo A: Planificación Carta Gantt.	60
Anexo A2: Github.	61
Anexo A3: Métricas de precisión, Recall, F1-Score.	62

Lista de Tablas

Tabla 3.1: Parámetros de los rodamientos (Fuente: [21]).	19
Tabla 3.2: Descripción de elementos utilizados para la medición. (Fuente: [21]).	20
Tabla 3.3: Estructura banco de datos (Fuente: [21]).	20
Tabla 3.4: Estructura de los bancos de imágenes. (Fuente: Elaboración propia.)	22
Tabla 3.5: Estructura de la CNN (Fuente: Elaboración propia.)	22
Tabla 3.6: Distribución de datos del banco de imágenes. (Fuente: Elaboración propia.)	23
Tabla 3.7: Modelos entrenados de CNN resultantes de la primera etapa. (Fuente: Elaboración propia.)	25
Tabla 3.8: Modelos entrenados de CNN para distintos anchos de ventana y normalizaciones para mejor duración de señal. (Fuente: Elaboración propia.)	25
Tabla 4.1: Resultados después de 10 iteraciones para cada caso usando datos de validación, 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.)	29
Tabla 4.2: Reporte de clasificación en datos de test para 10 giros del eje, normalización global, 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.)	32
Tabla 4.3: Resultados después de 10 iteraciones para cada caso usando datos de validación, 30 épocas y <i>learning rate</i> de 0,0001. (Fuente: Elaboración propia.)	34
Tabla 4.4: Reporte de clasificación en datos de test para mejores configuraciones. (Fuente: Elaboración propia.)	38
Tabla 5.1: Estructura banco de datos y casos de estudio. (Fuente: [27]).	48
Tabla 5.2: Parámetros de los rodamientos de banco de datos MFPT. (Fuente: [27]).	48
Tabla 5.3: Estructura de los bancos de imágenes MFPT. (Fuente: Elaboración propia.)	49
Tabla 5.4: Resultados después de 10 iteraciones para cada caso. (Fuente: Elaboración propia.)	55

Lista de Figuras

Figura 1.1: Número de publicaciones recientes aplicando DL para el diagnóstico de fallas en máquinas rotatorias. (Fuente: [5])	2
Figura 2.1: Estructura tradicional de una CNN. (Fuente: [5]).....	6
Figura 2.2: Operación de convolución. (Fuente: [15].).....	7
Figura 2.3: Funciones de activación ReLU, Sigmoid y Tanh. (Fuente: Elaboración propia.).....	8
Figura 2.4: Ejemplo de proceso de (a) MaxPooling y (b) AveragePooling. (Fuente:[16].).....	8
Figura 2.5: Espectrograma y escalograma para una señal de vibración. (Fuente: [19])	11
Figura 2.6: Ejemplo de una función de ventana. (Fuente: [18])	12
Figura 2.7: Función de ventana Hamming (Fuente: Elaboración propia.).....	12
Figura 2.8: Señales de vibración de rodamiento con distintos tipos de falla (Fuente: Elaboración propia.)	13
Figura 2.9: Imágenes tiempo-frecuencia para distintas señales de vibración y anchos de ventana. (Fuente: Elaboración propia.).....	14
Figura 2.10: Esquema generación imagen tiempo-frecuencia con normalización global (Fuente: Elaboración propia.)	15
Figura 2.11: Esquema para selección de amplitud máxima con normalización local. (Fuente: Elaboración propia.)	16
Figura 2.12: Esquema generación imagen tiempo-frecuencia con normalización por señal (Fuente: Elaboración propia.)	17
Figura 2.13: Imagen tiempo-frecuencia para normalización global, etiqueta: O-A-2. (Fuente: Elaboración propia.)	17
Figura 2.14: Imagen tiempo-frecuencia para normalización local, etiqueta: O-A-2. (Fuente: Elaboración propia.)	18
Figura 2.15: Imagen tiempo-frecuencia para normalización por señal, etiqueta: O-A-2. (Fuente: Elaboración propia.)	18
Figura 3.1: Banco de simulación de Fallas MFS-PK5M SpectraQuest (Fuente: [21]).....	19
Figura 3.2: Flujo general para la obtención de resultados. (Fuente: Elaboración propia.)	21
Figura 3.3: Proceso para generación de bancos de imágenes. (Fuente: Elaboración propia.)	21
Figura 3.4: Visualización gráfica de la arquitectura de la CNN (Fuente: Elaboración propia.)	23
Figura 3.5: Visualización gráfica de la arquitectura de la CNN (Fuente: Elaboración propia.)	24
Figura 4.1: Resultados de precisión promedio para duración de señal 0,6 segundos, 10 épocas y <i>learning rate</i> 0,001. (Fuente: Elaboración propia.).....	27
Figura 4.2: Resultados de precisión promedio duración de señal 1 segundo, 10 épocas y <i>learning rate</i> 0,001. (Fuente: Elaboración propia.).....	28
Figura 4.3: Resultados de precisión promedio para duración de señal correspondiente a 10 giros del eje, 10 épocas y <i>learning rate</i> 0,001. (Fuente: Elaboración propia.)	28
Figura 4.4: Resultados de precisión, pérdida y test para 10 giros del eje (mejores modelos), 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.).....	30

Figura 4.5: Matriz de confusión para datos de test para 10 giros del eje, 10 épocas y <i>learning rate</i> de 0,001 (mejores modelos). (Fuente: Elaboración propia.).....	31
Figura 4.6: Resultados de precisión promedio para duración de señal 0,6 segundos, 30 épocas y <i>learning rate</i> 0,0001. (Fuente: Elaboración propia.).....	33
Figura 4.7: Resultados de precisión promedio duración de señal 1 segundo, 30 épocas y <i>learning rate</i> de 0,0001. (Fuente: Elaboración propia.).....	33
Figura 4.8: Resultados de precisión promedio para duración de señal correspondiente a 10 giros del eje, 30 épocas y <i>learning rate</i> de 0,0001. (Fuente: Elaboración propia.)	34
Figura 4.9: Resultados de precisión, pérdida y test para 10 giros del eje (mejores modelos), 30 épocas y <i>learning rate</i> de 0,0001.	35
Figura 4.10: Resultados de precisión promedio para duración 10 giros del eje, normalización global, 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.).....	36
Figura 4.11: Resultados de precisión promedio para duración 10 giros del eje, normalización local, 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.).....	37
Figura 4.12: Resultados de precisión promedio para duración 10 giros del eje, normalización por señal, 10 épocas y <i>learning rate</i> de 0,001. (Fuente: Elaboración propia.).....	37
Figura 4.13: Resultados de datos de test normalización local, 10 giros del eje, ancho de ventana 4096 puntos de la señal. (Fuente: Elaboración propia.).....	38
Figura 4.14: Resultados de precisión promedio para duración 0,6 segundo y normalización local. (Fuente: Elaboración propia.).....	39
Figura 4.15: Resultados de precisión promedio para duración 1 segundo y normalización local. (Fuente: Elaboración propia.).....	40
Figura 4.16: Resultados de precisión promedio para duración 10 giros del eje y normalización por local. (Fuente: Elaboración propia.).....	40
Figura 4.17: Resultados t-SNE para 10 giros, normalización global y 4096 ancho de ventana. (Fuente: Elaboración propia.).....	41
Figura 4.18: Resultados t-SNE para 10 giros, normalización local y 4096 ancho de ventana. (Fuente: Elaboración propia.).....	42
Figura 4.19: Resultados t-SNE para 10 giros, normalización por señal y 4096 ancho de ventana. (Fuente: Elaboración propia.).....	42
Figura 4.20: Mapas de características para falla en bola, imagen: B-A-1_0. (Fuente: Elaboración propia.)	43
Figura 4.21: Mapas de características para falla combinada, imagen: C-A-1_0. (Fuente: Elaboración propia.)	44
Figura 4.22: Mapas de características para falla pista externa, imagen: O-A-1_0. (Fuente: Elaboración propia.)	44
Figura 4.23: Mapas de características para falla pista interna, imagen: I-A-2_0. (Fuente: Elaboración propia.)	45
Figura 4.24: Mapas de características para condición normal, imagen: H-A-1_0. (Fuente: Elaboración propia.)	45
Figura 4.25: Resultados de la regresión logística. (Fuente: Elaboración propia.).....	46

Figura 5.1: Ejemplos intuitivos de la Transferencia de conocimiento. (Fuente: [25]).....	47
Figura 5.2: Diagrama explicativo del concepto de <i>fine-tuning</i> . (Fuente: Elaboración propia).....	50
Figura 5.3: Gráficos de precisión y pérdida por épocas, (a) 3 giros del eje y (b) 10 giros del eje aplicando TL. (Fuente: Elaboración propia)	51
Figura 5.4: Matriz de confusión para datos de test, (a) 3 giros del eje y (b) 10 giros del eje aplicando TL. (Fuente: Elaboración propia).....	52
Figura 5.5: Gráficos de precisión y pérdida por épocas, (a) 3 giros del eje y (b) 10 giros del eje. (Fuente: Elaboración propia).....	53
Figura 5.6: Matriz de confusión para datos de test, (a) 3 giros del eje y (b) 10 giros del eje. (Fuente: Elaboración propia).....	54
Figura A3.1: Matriz de confusión para datos de test 10 giros del eje y normalización global. (Fuente: Elaboración propia).....	63

Acrónimos

DL	:	<i>Deep Learning.</i>
AE	:	<i>Autoencoders.</i>
RNN	:	<i>Recurrent Neural Networks.</i>
CNN	:	<i>Convolutional Neural Networks.</i>
STFT	:	<i>Short Time Fourier Transform.</i>
CWT	:	<i>Continuous Wavelet Transform.</i>
TL	:	<i>Transfer Learning.</i>
WVD	:	<i>Wigner-Ville Distribution.</i>
MFPT	:	<i>Society For Machinery Failure Prevention Technology.</i>
PHM	:	<i>Prognosis and Health Management.</i>
DBN	:	<i>Deep Belief Networks.</i>
GAN	:	<i>Generative Adversarial Network.</i>
PCA	:	<i>Principal Component Analysis.</i>

CAPÍTULO 1: Introducción

1.1 Planteamiento del problema

Los rodamientos son parte fundamental del funcionamiento de las máquinas rotatorias, las que se encuentran presentes en diversas industrias tales como manufactura, minería, celulosa, entre otras. De acuerdo con datos del *Electric Power Research Institute* los rodamientos acumulan un 41% de las fallas en máquinas rotatorias y se atribuyen un 40% de las fallas en motores [1,2]. La falla de este componente puede significar la detención completa de la línea de producción, pérdidas en seguridad, propiedad e incluso la pérdida de vidas en algunos casos [2]. Por lo que es de vital importancia contar con metodologías y técnicas que permitan monitorear el estado de salud de estos componentes sin tener la necesidad de detener la máquina.

La forma más habitual de prevenir estas fallas es a través de la implementación de un monitoreo de condición en tiempo real, que se realiza usualmente a través de la información obtenida de sensores ubicados en distintas partes de la máquina rotatoria. De los cuales, el sensor de vibraciones o acelerómetro corresponde al más utilizado para diagnosticar el estado de salud del rodamiento. Lo cual se realiza normalmente mediante técnicas de análisis de señales [2].

El análisis de las señales vibratorias se realiza empleando técnicas en el dominio del tiempo, frecuencia y tiempo-frecuencia. Esta última cuenta con la ventaja de tener la capacidad de analizar señales estacionarias o no estacionarias. Entre los métodos para analizar el dominio tiempo-frecuencia se encuentran la *Short Time Fourier Transform* (STFT), la *Wigner-Ville Distribution* (WVD), la *Continuous Wavelet Transform* (CWT), entre otras. La aplicación de estas transformadas genera una imagen de la distribución de energía de la señal vibratoria en los dominios del tiempo y frecuencia.

El desarrollo de la industria 4.0 trae consigo un gran aumento en la cantidad de datos de medición disponibles de las máquinas rotatorias y una mayor necesidad por la automatización del monitoreo de estas. Para dar solución a estas problemáticas se han desarrollado distintas metodologías dentro del aprendizaje automático y profundo, los que corresponden a subramas de la inteligencia artificial. Donde el aprendizaje profundo o *Deep Learning* (DL) ha ganado popularidad en estos últimos años.

Entre las principales ventajas del desarrollo de metodologías de DL es la capacidad de estas para realizar el monitoreo, diagnóstico y pronóstico con un alto nivel de automatización, requiriéndose poca intervención o necesidad de conocimiento de expertos [3]. Por lo que el uso de estas metodologías permite mejorar la eficiencia de la máquina rotatoria. Lográndose prolongar su vida útil, aumentar la confiabilidad y disminuir los costos de mantención [4].

1.2 Estado del arte

En el área correspondiente a *Prognosis and Health Management* (PHM), específicamente en lo que respecta al diagnóstico inteligente de máquinas rotatorias, se han desarrollado numerosas publicaciones basadas en metodologías de DL. En la Figura 1.1 se ilustran los modelos de DL más utilizados en estas publicaciones, en donde se encuentran las *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), *Autoencoders* (AE), *Deep Belief Networks* (DBN) y *Generative Adversarial Network* (GAN) [5].

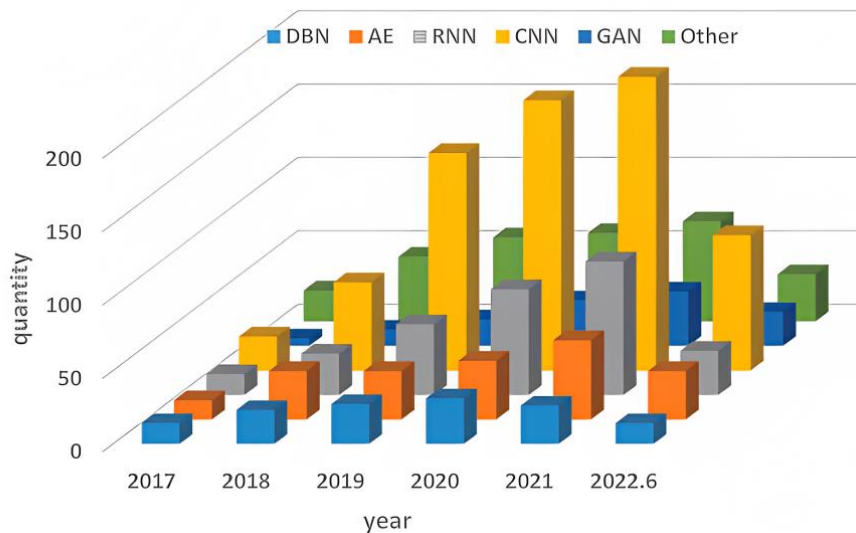


Figura 1.1: Número de publicaciones recientes aplicando DL para el diagnóstico de fallas en máquinas rotatorias. (Fuente: [5])

A continuación, se enumeran ciertos aportes realizados por autores aplicando modelos de DL para el diagnóstico de fallas en rodamientos. En [6] se presenta una metodología que utiliza la CWT para crear una representación tiempo-frecuencia de la señal de vibración, la cual es utilizada para el entrenamiento de la CNN. Esta arquitectura cuenta con una capa de convolución binaria local en reemplazo de la capa de convolución tradicionalmente presente en la CNN. Lo anterior, permite que esta arquitectura denominada CWT – LBCNN tenga una velocidad de entrenamiento más rápida y sea menos propensa al sobre ajuste. Los resultados que se obtienen con esta metodología son de un 95,29% de precisión en promedio para 16 clases, entre las que se consideran la condición normal y fallas en la pista interna o externa de distinta gravedad. Esta metodología se valida con el uso del banco de datos de rodamientos publicado por la *Society For Machinery Failure Prevention Technology* (MFPT).

La CNN como modelo de DL presenta su mayor potencial extrayendo características de señales en dos dimensiones (2-D). Sin embargo, existe una gran cantidad de publicaciones las cuales utilizan como entrada la señal de vibración sin ningún preprocesamiento y en una dimensión (1-D). Por ejemplo, en [7] se utiliza la señal de corriente del motor para diagnosticar la presencia de falla a través del uso de una CNN adaptativa, con tal de fusionar la extracción de características y clasificación de

clases. Se obtiene como resultado porcentajes de precisión del orden del 95-96% y se rescata la aplicabilidad de esta metodología para casos de monitoreo en tiempo real, dada la baja complejidad de la arquitectura propuesta.

En [8] se desarrolla una metodología enfocada en solucionar una de las principales problemáticas en cuanto a la generación de arquitecturas de modelos de DL. La que se encuentra marcada por la prueba y error, además de la falta de una guía teórica para construir estas estructuras. Para lo anterior se propone el uso de *Principal Component Analysis* sobre la señal de vibración de entrada, disminuyendo su dimensionalidad. Luego se utiliza un algoritmo evolutivo para generar la estructura óptima de la DBN a medida que se entrena. Los resultados de esta metodología alcanzan un 99,73 % de precisión para 6 modos de fallas de distinta gravedad y clase.

Entre otras aplicaciones, en [9] se utiliza un método para generar una mezcla de datos simulados y reales, con tal de solucionar la falta de datos de falla para algún modo de falla en particular. Para esto se utiliza un modelo dinámico de un rodamiento para obtener datos de simulación. Luego, el modelo es entrenado con los datos simulados para distintas condiciones y son utilizados para convertir datos reales a las nuevas condiciones. Finalmente, con el conjunto de los datos generados se entrena una GAN. Este método resuelve el problema de la falta de datos para nuevas condiciones, alcanzando una precisión un 99,60% para 5 modos de falla.

A diferencia de los aportes que se mencionan anteriormente, los que suelen utilizar como base modelos de DL con arquitecturas tradicionales. En [10] se propone el uso de un modelo *Transformer* cuya publicación se realiza en 2017 [11], este modelo presenta aplicaciones en el ámbito del procesamiento de lenguaje natural, imágenes y últimamente en el diagnóstico de fallas. La metodología propuesta es similar en cuanto al preprocesamiento de los datos de vibración, en donde se tiene como entrada una imagen tiempo-frecuencia derivada del uso de la Transformada Sincronizada Wavelet. Luego se realiza la fase de entrenamiento del modelo para que este aprenda las características o atributos de la imagen tiempo-frecuencia, con tal de realizar la tarea de clasificación. El desempeño del modelo se evalúa con respecto a los datos recolectados de un banco de ensayos, obteniéndose como resultado un 99,94% de precisión promedio para el modelo propuesto y un 92,56% para una CNN entrenada sobre el mismo caso.

La ventaja principal del modelo *Transformer* por sobre modelos tradicionales como la CNN y RNN, es su capacidad de ejecución de procesos en paralelo y la disminución considerable de parámetros a aprender por el modelo, lo cual significa que su tiempo de entrenamiento sea el menor dentro de los modelos de DL. Además, supera las dificultades inherentes de la naturaleza de los modelos CNN y RNN. Lo que significa para el caso de la RNN, que su arquitectura no sea ideal para la computación de procesos en paralelo. Por lo que es ineficiente en el entrenamiento de banco de datos de gran escala. Además, no se puede desprender del problema de establecer relaciones o conexiones efectivas entre secuencias distantes. Por otra parte, la CNN sufre de pequeñas desventajas tales como la necesidad de considerar varias capas convolucionales dado el campo de recepción determinado por el tamaño del kernel a utilizar, aumentando la cantidad de parámetros a entrenar.

Actualmente, el foco de investigación según [12] se encuentra en aumentar la capacidad de generalización de estos modelos. Además de abarcar la problemática que presenta la distribución

desigual entre los datos de entrenamiento y test en la industria, dada la poca información acerca de sucesos de fallas. A estas razones, la Transferencia de Conocimiento o *Transfer Learning* (TL) surge como solución para aliviar parte de estas problemáticas. El TL consiste principalmente en aplicar el conocimiento aprendido en un dominio fuente y aplicarlo a otro dominio distinto.

Para ejemplificar el proceso de TL considerar un modelo de DL, el cual es entrenado para reconocer fallas en un tipo de rodamiento y con ciertas condiciones de operación. Posteriormente, se necesita aplicar la misma metodología para otro equipo que presenta un rodamiento distinto y con otras condiciones de operación, del cual se dispone de pocos datos para entrenar un modelo de DL desde cero. A lo cual entra en contexto el uso de TL, con tal de aplicar lo aprendido en el dominio fuente y mediante distintas técnicas como el *fine-tuning* adaptarlo a las nuevas condiciones. Lográndose disminuir los tiempos de entrenamiento y aumentar la capacidad de generalización del modelo, ya que se dispone de un modelo capaz de diagnosticar la fallas sin la necesidad de un gran número de datos de fallas.

A pesar de la gran cantidad de metodologías disponibles para el diagnóstico de fallas en rodamientos aplicando modelos de DL. La gran mayoría de esta no considera la evaluación de distintas configuraciones en el preprocesamiento de los datos previos al entrenamiento del modelo de DL, por lo que entre los aportes de esta memoria se encuentra la evaluación de la influencia de distintas configuraciones para la generación de las imágenes tiempo-frecuencia. Además, de proponer una metodología con una mayor capacidad de generalización, evaluándose así la posibilidad de desarrollar TL para este propósito.

1.3 Hipótesis

El modelo de aprendizaje profundo clasifica de manera correcta la falla presente en el rodamiento mediante las imágenes tiempo-frecuencia generadas por la STFT.

1.4 Metodología

La metodología se basa principalmente en la aplicación de técnicas de preprocesamiento de datos y de transformadas tiempo-frecuencia para un banco de datos de rodamientos extraído de la literatura. De lo anterior, se generan distintos bancos de imágenes que son utilizados para el entrenamiento de una arquitectura de la CNN. Lo que se prosigue con un análisis de sensibilidad de los resultados para las distintas configuraciones consideradas para la generación de estos bancos de imágenes mediante la STFT.

Finalmente, se aplica TL a otro banco de datos de rodamientos y se evalúa la factibilidad de este proceso con base en los resultados obtenidos.

El entrenamiento es realizado en Python 3.11.3, y un computador con las siguientes especificaciones en hardware: Ryzen 5600x CPU, RTX 3060 GPU, 16 GB RAM.

1.5 Objetivo general

Generar modelos de aprendizaje profundo para la clasificación automática de imágenes tiempo-frecuencia asociadas a fallas en rodamientos.

1.6 Objetivos específicos

1. **OE1:** Investigación bibliográfica acerca de modelos de aprendizaje profundo usados para el diagnóstico de fallas en rodamientos.
2. **OE2:** Generar bancos de datos de imágenes tiempo-frecuencia etiquetadas según el tipo de falla que presenta utilizando distintas técnicas de preprocesamiento de datos.
3. **OE3:** Entrenar modelos de aprendizaje profundo que en base a las imágenes tiempo-frecuencia clasifique el tipo de falla que presenta el rodamiento.
4. **OE4:** Realizar análisis de sensibilidad de los resultados obtenidos por los modelos de aprendizaje profundo.
5. **OE5:** Evaluar la factibilidad de realizar transferencia de conocimiento a otro banco de datos de rodamientos.

CAPÍTULO 2: Marco teórico

En este capítulo se presentan los fundamentos teóricos detrás del funcionamiento de la CNN y de las transformadas Tiempo-Frecuencia, haciéndose especial énfasis en la STFT y sus parámetros.

2.1 Redes Neuronales Convolucionales.

La CNN corresponden a un modelo de DL el cual se basa en imitar el mecanismo de la visión de los animales. La primera arquitectura propuesta corresponde a LeNet por LeGun en 1989 [13], dada la gran capacidad de este modelo para procesar imágenes se han desarrollado distintas arquitecturas. Entre las cuales destacan AlexNet, VGGNet, ResNet y MobileNet [5].

La estructura tradicional de funcionamiento de las CNN se compone principalmente de capas convolucionales, capas de Pooling y capas de neuronas completamente conectadas. En la Figura 2.1 se presentan un ejemplo de los componentes que constituyen una CNN.

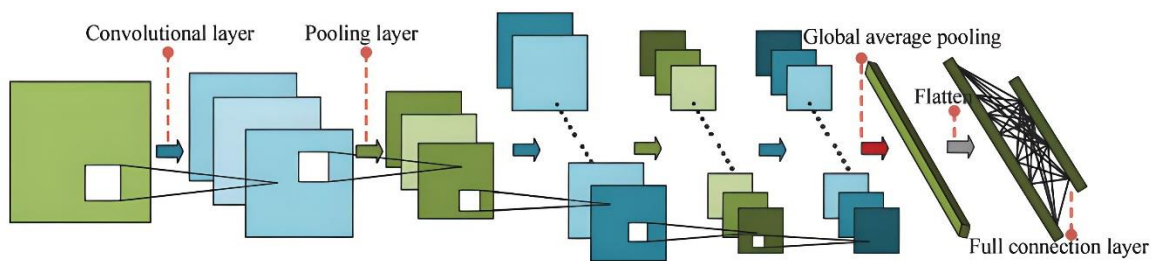


Figura 2.1: Estructura tradicional de una CNN. (Fuente: [5])

2.1.1 Capa convolucional.

La capa de convolución es la encargada de multiplicar el kernel o filtro con una parte de la imagen, luego se suman los números resultantes y se asignan a una nueva matriz. Este procedimiento se realiza a lo largo de toda la imagen, lo que en conjunto con el uso de distintos tipos de kernel deriva en la generación de los mapas de características. Los que contienen información relevante acerca de la imagen de entrada.

La capa de convolución y la capa previa se encuentran conectadas a través de la conexión local de sus neuronas y pesos compartidos, lo que reduce considerablemente la cantidad de parámetros de la CNN [14].

Esta operación se define matemáticamente como:

$$h_j = f\left(\sum_i X_i * W_{ij} + b_j\right) \quad (1)$$

Donde h_j indica el mapa de características de salida de la presente capa convolucional, X_i representa el mapa de características de la capa previa. La operación de convolución se denota por $*$, W_{ij} es la matriz de peso del kernel de convolución, b_j es el término offset correspondiente a cada característica de la capa de convolución y $f()$ corresponde a la función de activación.

En la Figura 2.2, se ilustra el proceso que ocurre en la capa de convolución.

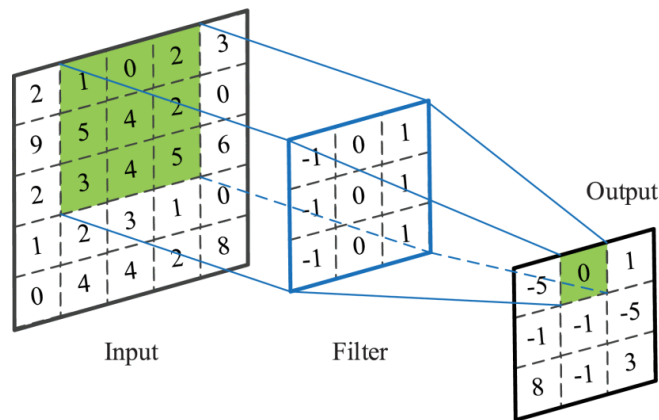


Figura 2.2: Operación de convolución. (Fuente: [15].)

2.1.2 Función de activación.

Para mejorar la capacidad de clasificación del modelo, se hace uso de una función de activación no-linear. Entre las más usadas comúnmente se encuentran la función ReLU, tanh y sigmoid [16]. De las cuales la función ReLU será utilizada para el entrenamiento de la CNN.

La función ReLU se define como:

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2)$$

La función tanh se define como:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

La función sigmoid se define como:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

En la Figura 2.3, se presenta en un gráfico la forma de las funciones anteriormente definidas.

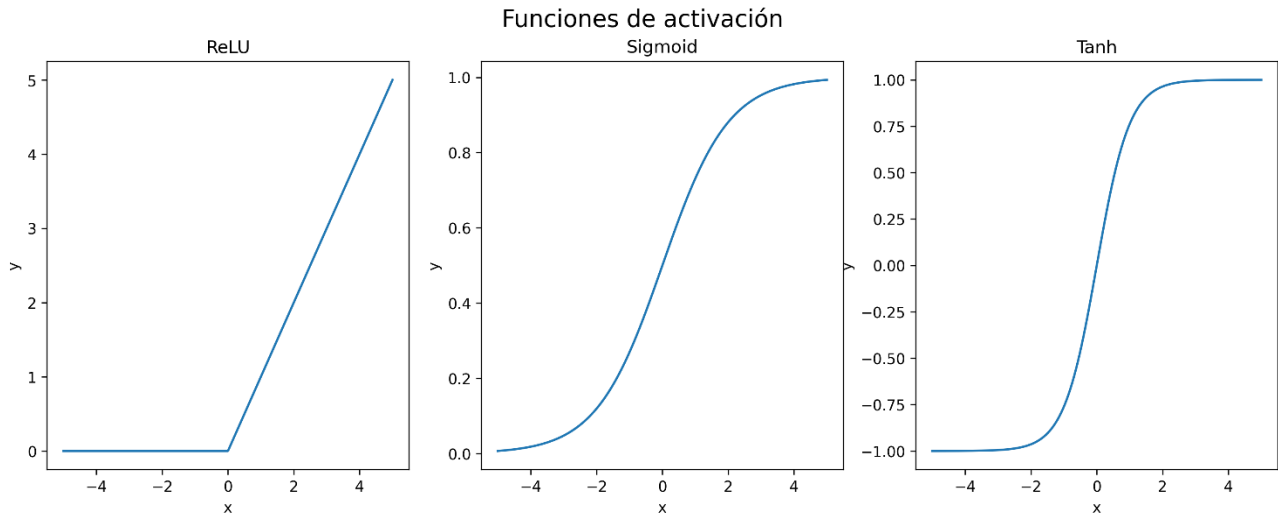


Figura 2.3: Funciones de activación ReLU, Sigmoid y Tanh. (Fuente: Elaboración propia.)

2.1.3 Capa de Pooling (Pooling Layer)

Si la capa de convolución tiene el rol de la extracción de características, la capa de Pooling cumple con seleccionar las características más representativas. Además, en este proceso se reduce la dimensión de la representación. De tal manera que se disminuye la cantidad de parámetros a entrenar por el modelo [16].

En la Figura 2.4, se ilustra el proceso de *MaxPooling* y *AveragePooling* que son las funciones de submuestreo más utilizadas [16]. De las cuales la función *MaxPooling* corresponde a la utilizada en el entrenamiento de la CNN.

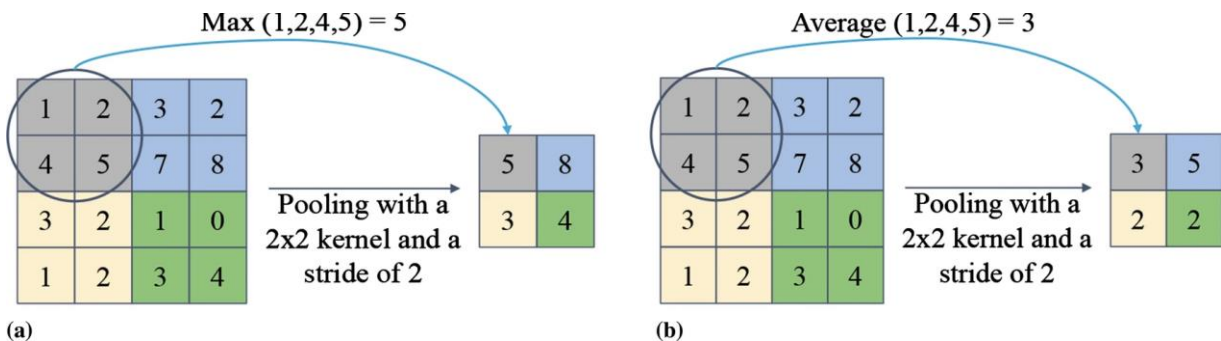


Figura 2.4: Ejemplo de proceso de (a) MaxPooling y (b) AveragePooling. (Fuente:[16].)

Matemáticamente estas operaciones se definen como:

$$H_j^{l+1} = f(\beta_j^{l+1} \text{down}(H_j^l) + b_j^l) \quad (5)$$

Donde H_j^{l+1} denota el j -th mapa de características en la capa $l + 1$, $\text{down}(\cdot)$ denota la función de Pooling y β_j^{l+1} representa el *bias* multiplicativo.

2.1.4 Capa completamente conectada y clasificador Softmax.

La capa completamente conectada es usada para clasificar las características extraídas de las imágenes en las capas de convolución y Pooling. Esta operación se expresa como:

$$y^k = f(w^k x^{k-1} + b^k) \quad (6)$$

Donde k es la k -th capa del modelo, x^{k-1} es la entrada $(k-1)$ -th de la capa completamente conectada. El y^k es la salida de la capa completamente conectada, w^k es el peso, b^k el *bias* y $f(\cdot)$ es la función de clasificación.

La función de clasificación comúnmente utilizada para tareas de clasificación múltiple es Softmax. Esta función mapea la salida de las múltiples neuronas con valores entre 0 y 1 [16], la expresión matemática que describe esta operación es:

$$q(x_i) = \frac{e^{x_c}}{\sum_{c=1}^C e^{x_c}} \quad (7)$$

Donde x corresponde al valor de entrada la capa Softmax, $q(x_i)$ es el vector de probabilidad con dimensión C .

2.1.5 Gradiente descendente y función de costo.

El proceso de entrenamiento de la CNN es similar al *Backpropagation* que comúnmente se utiliza en las redes neuronales, con el objetivo de minimizar el error. Este error corresponde a la diferencia entre el valor de salida y_{ik} y el valor predicho d_{ik} [17]. Entre las funciones de costo se encuentran varios tipos, a continuación se presenta la función *mean-square error* (MSE):

$$E = \frac{1}{2} \sum_{k=1}^m (y_{ik}^n - d_{ik}^n)^2 \quad (8)$$

El método de gradiente descendente se define como:

$$\begin{cases} \omega' = \omega - \alpha \frac{\partial E}{\partial \omega} \\ b' = b - \alpha \frac{\partial E}{\partial b} \end{cases} \quad (9)$$

Donde ω' y ω es el peso actual y de la iteración anterior respectivamente; b' y b son el *bias* de la iteración actual y anterior. La velocidad de aprendizaje se denota como α y E es la función de costo.

2.2 Transformadas Tiempo – Frecuencia.

Las señales en el dominio tiempo y frecuencia son usadas comúnmente para monitorear el estado de salud de una máquina rotatoria. Sin embargo, estos dominios por separado no pueden describir los cambios en la señal correspondientes a la relación entre el tiempo y frecuencia de la señal en su conjunto. Esto se presenta como una desventaja dada la gran cantidad aplicaciones que cuentan con un comportamiento no-estacionario, tales como la señal de un radar, ondas sísmicas y señales de vibración [15].

Las transformadas Tiempo-Frecuencia (T-F) cuentan con la ventaja de poder analizar las señales de vibración independiente del régimen en el cual se encuentre operando la máquina, preservando la información temporal. Estas herramientas entregan como resultante una imagen completa del dominio T-F de la señal a analizar, por lo que es crucial la correcta selección de la Transformada T-F y sus parámetros. De manera que esta imagen contenga información relevante sobre el estado del componente, con la finalidad de entrenar un modelo de DL [18]. En esta imagen resultante se identifican no solo las frecuencias que se encuentran presentes en la señal, sino que también el cómo evolucionan con el tiempo.

Entre las representaciones más comunes se encuentran el espectrograma y escalograma. El espectrograma corresponde a la resultante de utilizar la STFT y el escalograma se refiere a la imagen resultante de aplicar la Transformada Wavelet. La principal diferencia entre estas dos técnicas es que el espectrograma depende del tamaño de la ventana y el escalograma de la frecuencia de la señal [19]. En la Figura 2.5, se presentan ejemplos de estas representaciones.

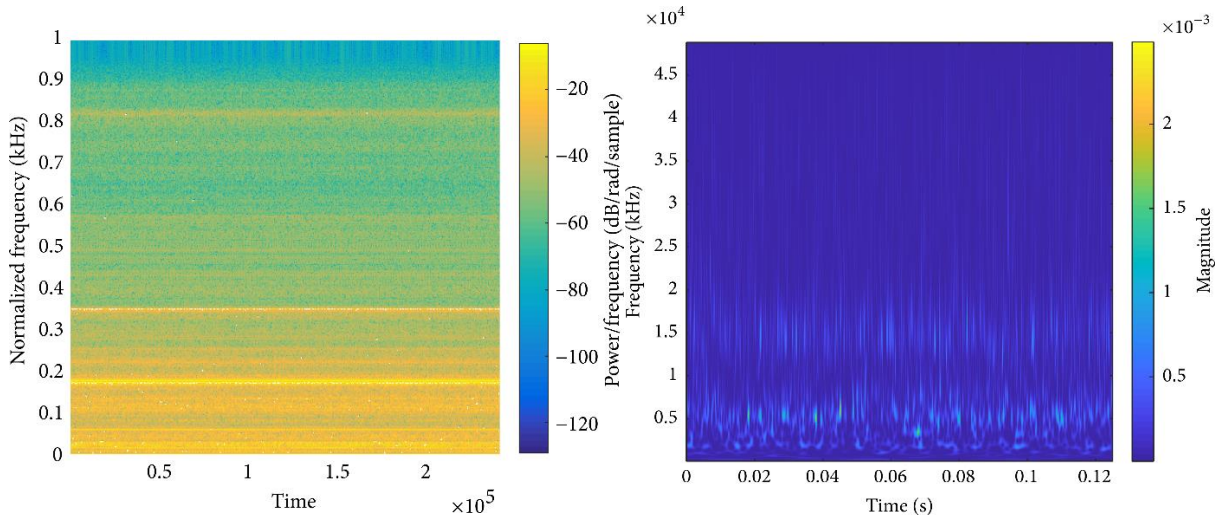


Figura 2.5: Espectrograma y escalograma para una señal de vibración. (Fuente: [19])

2.3 Transformada Corta de Fourier

La Transformada Corta de Fourier fue la primera transformada T-F aplicada en el campo de la comunicación por voz en 1946 por Gabor [18]. El principio de funcionamiento de la STFT consiste en dividir la señal no-estacionaria en pequeños segmentos, los cuales se asumen localmente estacionarios. Posteriormente, se aplica la Transformada de Fourier a cada uno de estos segmentos y el conjunto de este análisis espectral conforma entonces la imagen T-F o espectrograma.

La STFT se calcula como:

$$STFT(t, \omega) = \int_{-\infty}^{\infty} s(\tau)g(\tau - t)e^{-j\omega t}ds \quad (10)$$

El espectrograma se calcula como:

$$|STFT(t, \omega)|^2 = \left| \int_{-\infty}^{\infty} s(\tau)g(\tau - t)e^{-j\omega t}ds \right|^2 = G(t, \omega) \quad (11)$$

En donde $s(\tau)$ es la señal de vibración, $g(t)$ es la función de ventana, t y t corresponden al momento, $g(\tau - t)$ denota la función de ventana centrada en el momento t y ω es la frecuencia.

La resolución en tiempo y frecuencia se determinan en base a la longitud de la ventana a considerar. En el caso de ser necesaria una mayor resolución el dominio frecuencia, se utiliza una ventana más larga. Por otra parte, para aumentar la frecuencia en el dominio tiempo, el ancho de ventana debe disminuir. Lo anterior corresponde a un intercambio entre la resolución del dominio tiempo y frecuencia, producto del Principio de incertidumbre. En la Figura 2.6, se muestra un ejemplo de función de ventana aplicada en una parte de la señal.

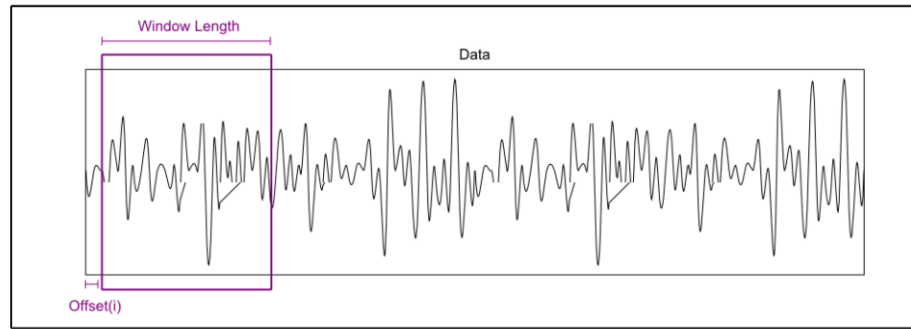


Figura 2.6: Ejemplo de una función de ventana. (Fuente: [18])

2.3.1 Función de ventana.

En este trabajo se considera la función de ventana Hamming para la generación de las representaciones tiempo-frecuencia. Entre las ventajas de esta función se encuentran la reducción de interferencias y aumento de la resolución espectral en la STFT. Por otra parte, el uso de esta ventana tiene como consecuencia la suavización de los componentes espectrales [20]. En la Figura 2.7, se visualiza la forma de la ventana Hamming en dominio tiempo y frecuencia.

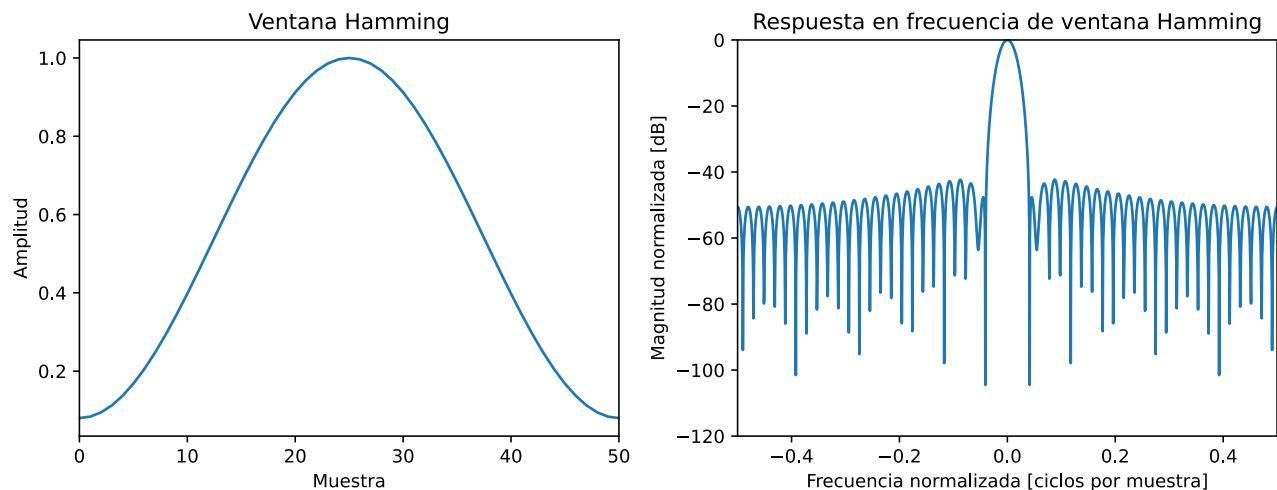


Figura 2.7: Función de ventana Hamming (Fuente: Elaboración propia.)

2.3.2 Ancho de ventana.

Para frecuencias bajas por lo general se utilizan ventanas de tamaño más grandes para aumentar la resolución en frecuencia. Para frecuencias más altas, se prefiere el uso de una ventana más pequeña con tal de aumentar la resolución en tiempo [19].

Para ilustrar la influencia de la longitud de ventana, en la Figura 2.9 se presentan los espectrogramas correspondientes a 0,2 segundos de una señal respectiva de la Figura 2.8. En donde se destaca la influencia de la longitud de ventana en la generación de la imagen y la importancia de evaluar el cómo afectaría esta configuración al desempeño del modelo a entrenar.

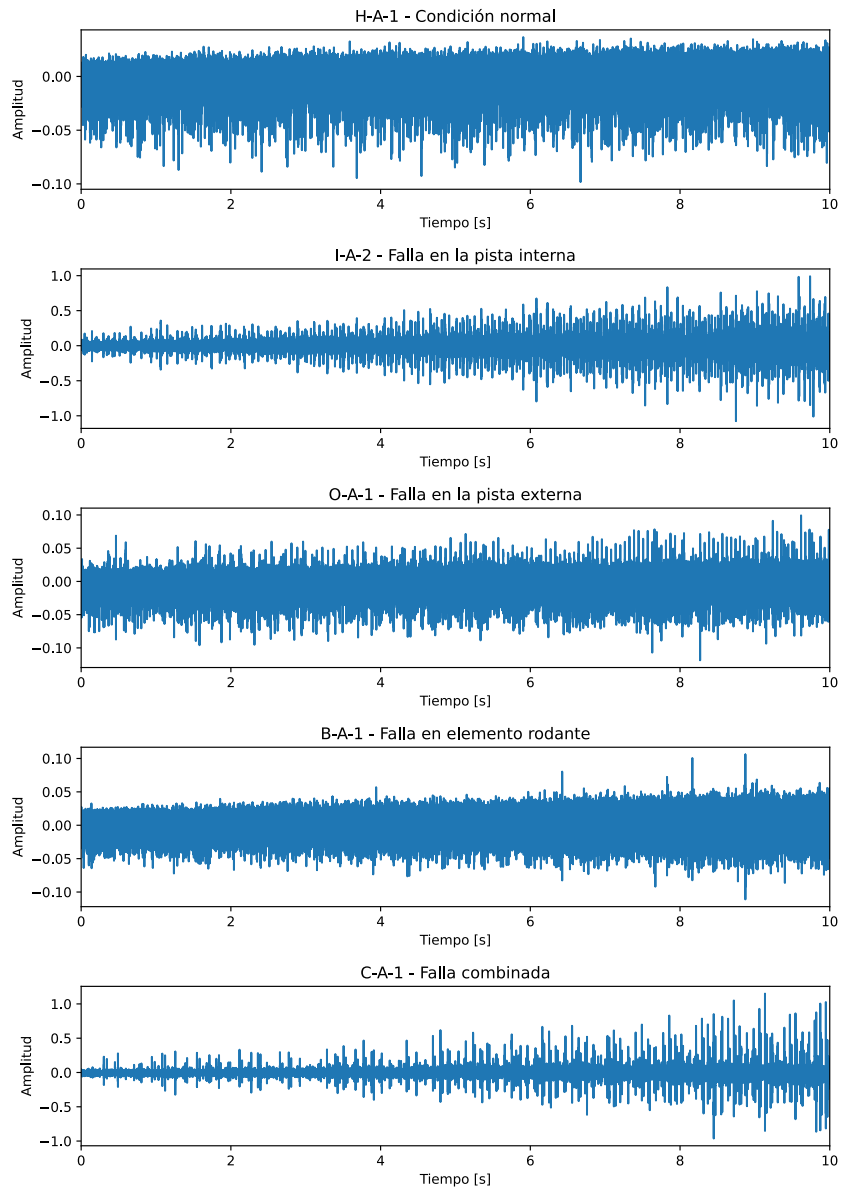


Figura 2.8: Señales de vibración de rodamiento con distintos tipos de falla (Fuente: Elaboración propia.)

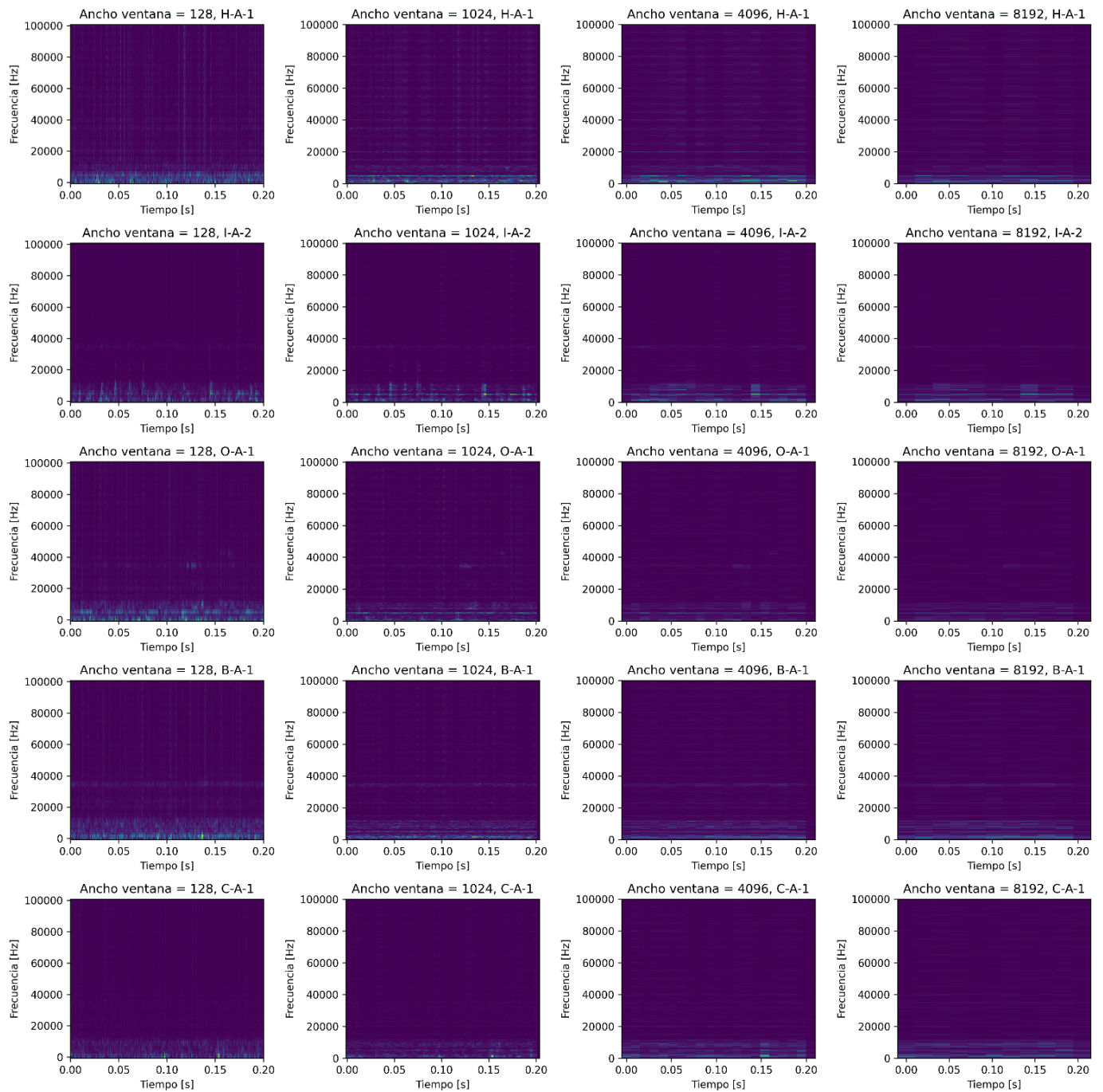


Figura 2.9: Imágenes tiempo-frecuencia para distintas señales de vibración y anchos de ventana. (Fuente: Elaboración propia.)

En la Figura 2.9, se visualiza el intercambio en resolución que se produce al utilizar una ventana pequeña para mayor resolución en el tiempo o una ventana más grande aumentando la resolución en la frecuencia.

2.4 Normalización

En la generación de la imagen T-F la intensidad de los píxeles en la escala de colores también afecta la calidad de la información que es capaz de extraer el modelo de DL, con tal de caracterizar y aprender las características relevantes del tipo de falla presente en la imagen T-F. Por lo que en este trabajo con tal de evaluar la mejor técnica de normalización en este aspecto. Se definen tres tipos de normalización a utilizar en la metodología de trabajo para la generación de los bancos de imágenes.

2.4.1 Normalización global

La normalización global consiste en escalar la intensidad de los colores de la imagen generada en base al banco de datos de estudio. En donde el valor de la máxima amplitud o intensidad del píxel corresponde al archivo que posee este valor.

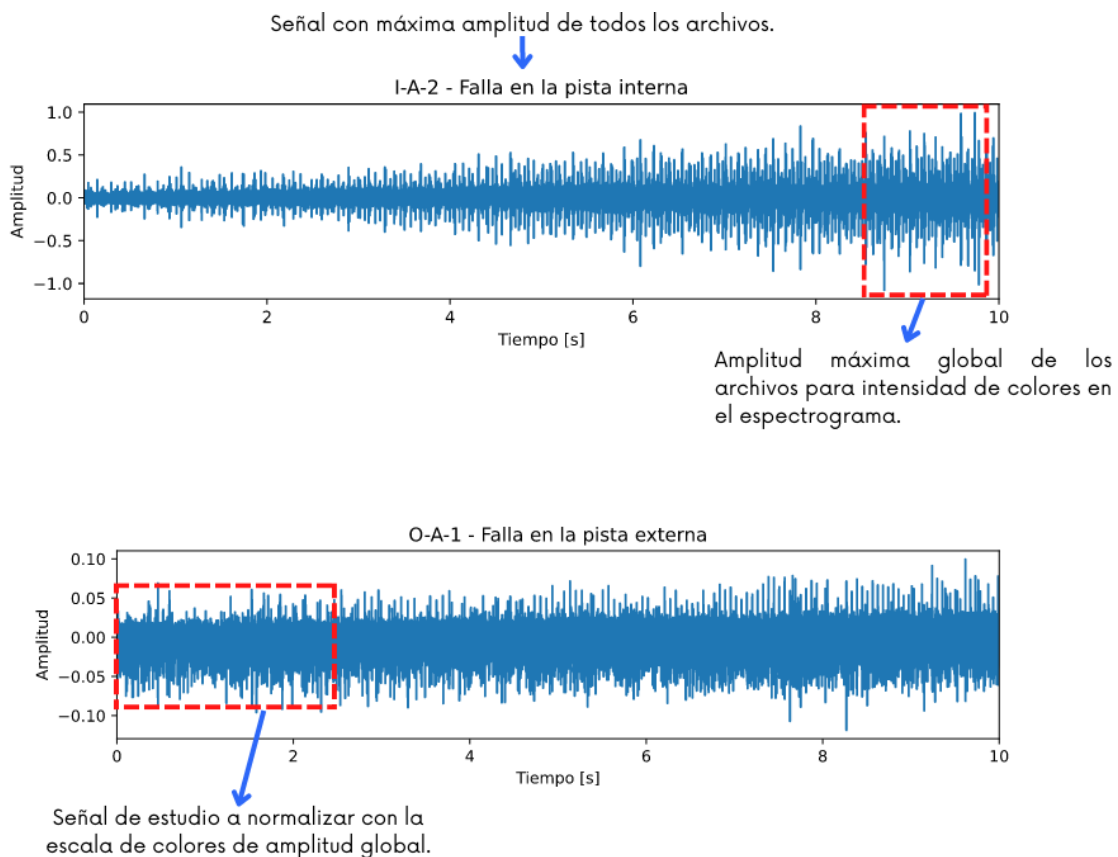


Figura 2.10: Esquema generación imagen tiempo-frecuencia con normalización global (Fuente: Elaboración propia.)

2.4.2 Normalización local

La normalización local consiste en escalar la intensidad de los colores con respecto a la máxima amplitud registrada en la ventana de señal a considerar para cada tramo por separado. En la Figura 2.11, se representa un esquema para la generación de esta imagen normalizada localmente.

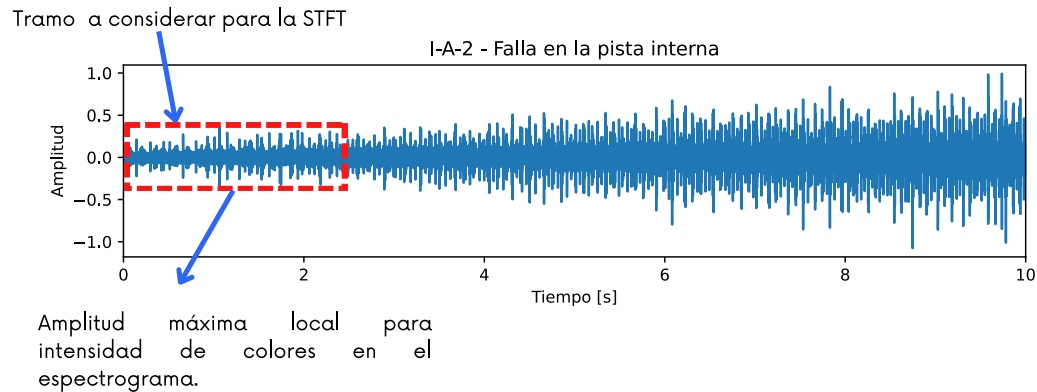


Figura 2.11: Esquema para selección de amplitud máxima con normalización local. (Fuente: Elaboración propia.)

2.4.3 Normalización por señal

En la normalización por señal a diferencia de las anteriores, se considera la intensidad de color registrada a lo largo de toda la señal de vibración presente en el archivo. En la Figura 2.12, se presenta un esquema para este proceso.

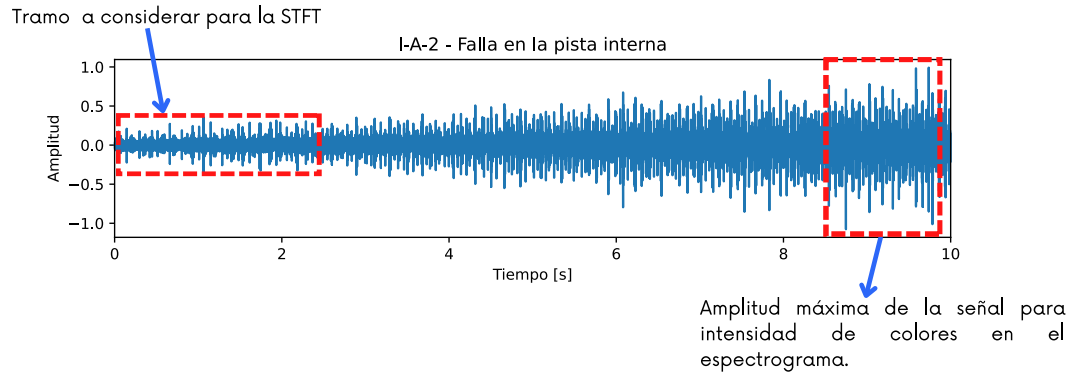


Figura 2.12: Esquema generación imagen tiempo-frecuencia con normalización por señal (Fuente: Elaboración propia.)

La diferencia entre estas distintas normalizaciones se puede observar en las Figuras 2.13, 2.14 y 2.15; en donde se distingue claramente la variación en la intensidad de los colores entre una normalización y otra.

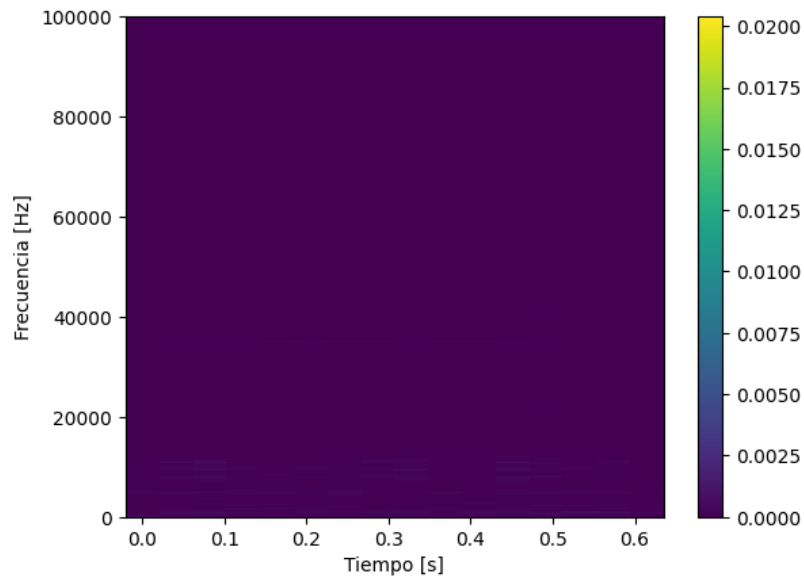


Figura 2.13: Imagen tiempo-frecuencia para normalización global, etiqueta: O-A-2. (Fuente: Elaboración propia.)

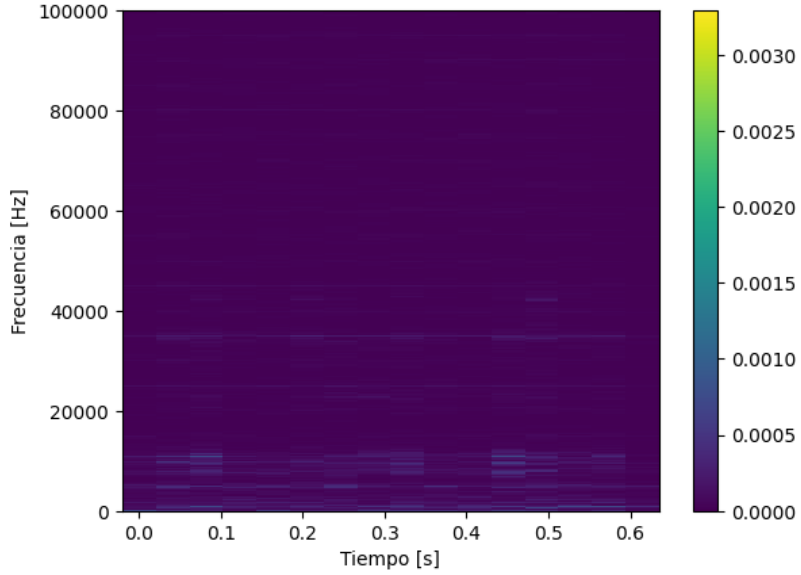


Figura 2.14: Imagen tiempo-frecuencia para normalización local, etiqueta: O-A-2. (Fuente: Elaboración propia.)

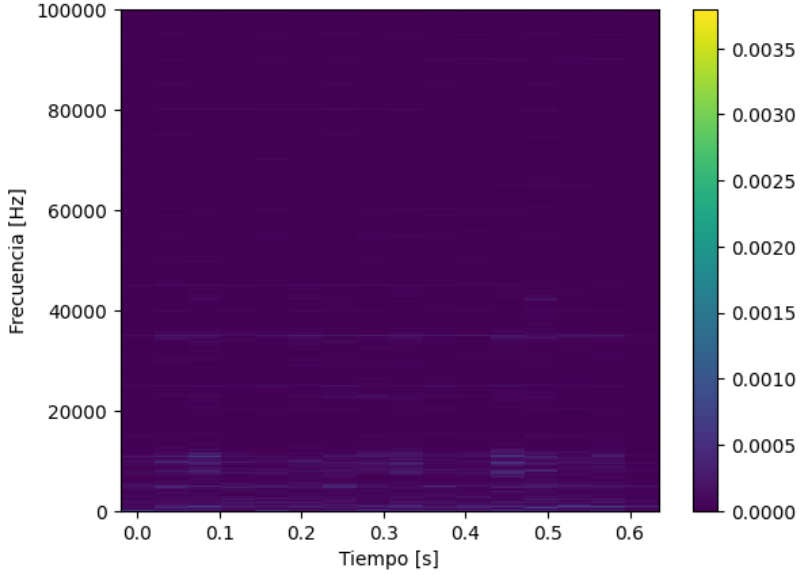


Figura 2.15: Imagen tiempo-frecuencia para normalización por señal, etiqueta: O-A-2. (Fuente: Elaboración propia.)

CAPÍTULO 3: Casos de estudio y metodología de trabajo.

3.1 Banco de simulación de fallas.

En este trabajo se utilizan datos de vibración de rodamientos con distintos tipos de fallas y variaciones en la velocidad del eje. Estos datos fueron recolectados por investigadores de la Universidad de Ottawa, Canadá [21].

En la Figura 3.1, se muestra el banco de simulación de fallas utilizado para la medición de los datos de vibración. El estudio se compone dos rodamientos de bolas EK16K, de los cuales el de la derecha presenta los distintos modos de falla.

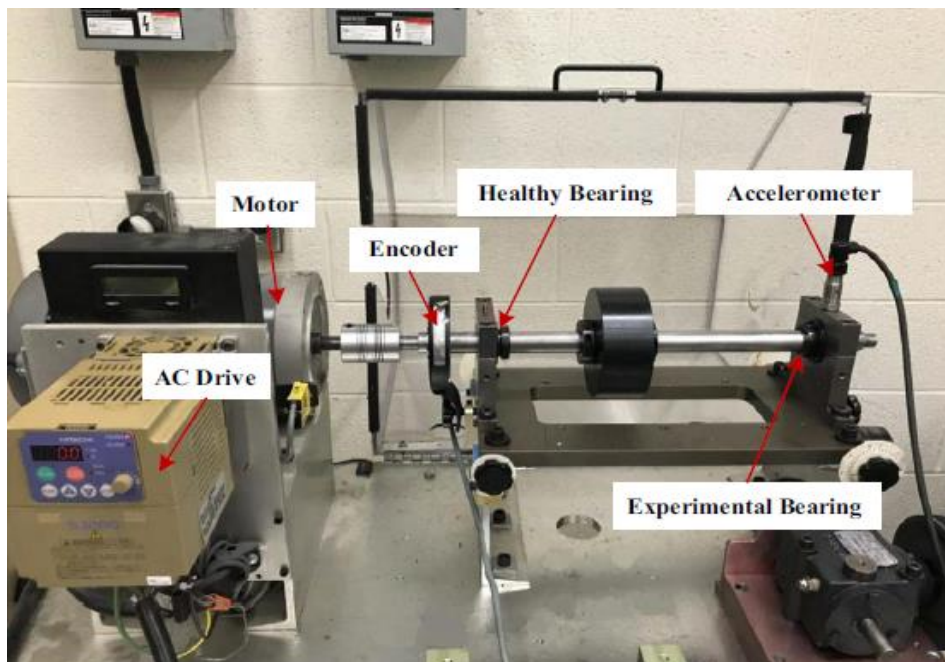


Figura 3.1: Banco de simulación de Fallas MFS-PK5M SpectraQuest (Fuente: [21])

Las características de los rodamientos se presentan en la Tabla 3.1, donde BPF_I y BPF_O son las frecuencias características de falla del rodamiento. Lo anterior, para la pista interna y externa respectivamente. Donde f_r es la velocidad de rotación.

Tabla 3.1: Parámetros de los rodamientos (Fuente: [21]).

Modelo rodamiento	Diámetro Pitch (mm)	Diámetro bolas (mm)	Número de bolas	BPF _I	BPF _O
ER16K	38.52	7.94	9	$5.43f_r$	$3.57f_r$

3.1.1 Banco de datos.

En la Tabla 3.2, se detallan los instrumentos y sensores que se utilizaron para la conformación del banco de datos de estudios referenciado anteriormente. Las mediciones se realizan para una frecuencia de muestreo de 200 kHz y una duración de 10 s.

Tabla 3.2: Descripción de elementos utilizados para la medición. (Fuente: [21]).

Instrumento o sensor	Descripción
NI USB-6212 BNC	Placa de adquisición de datos, guarda mediciones en un archivo (.mat), en donde el 'Channel_1' registra las mediciones del acelerómetro y el 'Channel_2' registra las mediciones del encoder.
Acelerómetro ICP, modelo 623C01	Sensor encargado de medir las vibraciones.
EPC modelo 775	Recolecta los pulsos correspondientes a los giros del eje.

El banco de datos consta de 60 archivos cuya estructura se presenta en la Tabla 3.3. Para distintas condiciones del rodamiento y cambios en la velocidad del eje.

Tabla 3.3: Estructura banco de datos (Fuente: [21]).

Condición del rodamiento	Aumentando velocidad	Disminuyendo velocidad	Aumentando y luego disminuyendo	Disminuyendo y luego aumentando.
Normal	H-A-1	H-B-1	H-C-1	H-D-1
	H-A-2	H-B-2	H-C-2	H-D-2
	H-A-3	H-B-3	H-C-3	H-D-3
Falla en la pista interna	I-A-1	I-B-1	I-C-1	I-D-1
	I-A-2	I-B-2	I-C-2	I-D-2
	I-A-3	I-B-3	I-C-3	I-D-3
Falla en la pista externa	O-A-1	O-B-1	O-C-1	O-D-1
	O-A-2	O-B-2	O-C-2	O-D-2
	O-A-3	O-B-3	O-C-3	O-D-3
Falla el elemento rodante	B-A-1	B-B-1	B-C-1	B-D-1
	B-A-2	B-B-2	B-C-2	B-D-2
	B-A-3	B-B-3	B-C-3	B-D-3
Falla combinada	C-A-1	C-B-1	C-C-1	C-D-1
	C-A-2	C-B-2	C-C-2	C-D-2
	C-A-3	C-B-3	C-C-3	C-D-3

3.2 Metodología de trabajo.

La metodología de trabajo se divide principalmente en cuatro tareas: (a) Generación de los bancos de imágenes; (b) Evaluación de mejores configuraciones: normalización, ancho de ventana y duración de señal; (c) Selección de modelo en base a mejores configuraciones; (d) Transferencia de conocimiento.

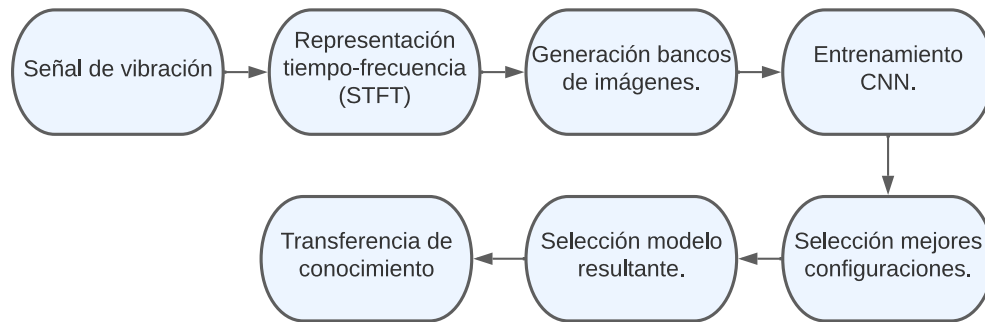


Figura 3.2: Flujo general para la obtención de resultados. (Fuente: Elaboración propia.)

3.2.1 Generación de bancos de imágenes.

A continuación, se detalla la metodología de trabajo para la generación de los bancos de imágenes.

1. Selección de duración de la señal.
2. Transformación de la señal de vibración a una imagen tiempo-frecuencia mediante la STFT para un ancho de ventana arbitrario de 16384 puntos de la señal.
3. Normalización de la escala de colores de la imagen para distintas amplitudes: local, global y por señal.
4. Almacenamiento de la imagen con etiqueta y posterior generación de banco de imágenes.

El detalle de los pasos se representa en la Figura 3.3, como resultado en una primera etapa se generan 9 bancos de imágenes cuya estructura se detalla en la Tabla 3.4. Donde las 5 clases corresponden al rodamiento con falla en la pista interna, externa, elemento rodante, combinada o rodamiento en condición normal.

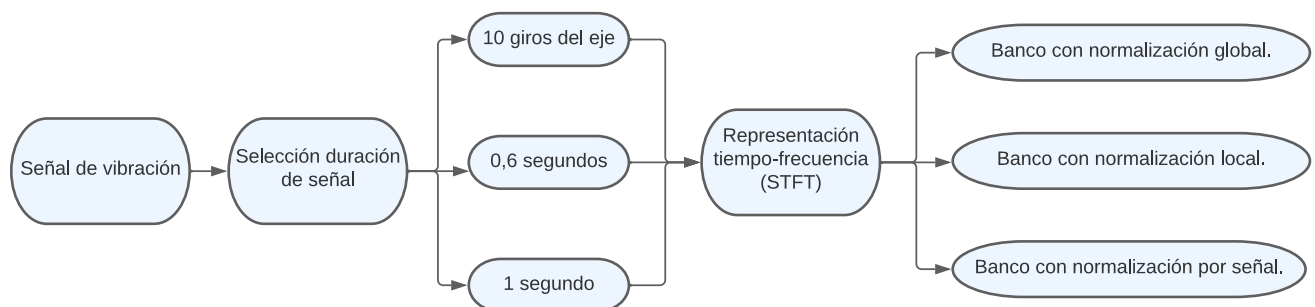


Figura 3.3: Proceso para generación de bancos de imágenes. (Fuente: Elaboración propia.)

Tabla 3.4: Estructura de los bancos de imágenes. (Fuente: Elaboración propia.).

Duración de señal	Normalización	Número de modos de falla.	Cantidad total de imágenes generadas
0,6 segundos	Global	5	944
	Local	5	944
	Por señal	5	944
1 segundo	Global	5	590
	Local	5	590
	Por señal	5	590
10 giros del eje	Global	5	1086
	Local	5	1086
	Por señal	5	1086

3.2.2 Arquitectura y entrenamiento de la CNN.

El entrenamiento de la CNN se realiza en base a una arquitectura fija inspirada en LeNet y el detalle de su estructura se presenta en la Tabla 3.5 y Figura 3.4. La función de activación es ReLU y el método de Pooling es *MaxPooling*.

Tabla 3.5: Estructura de la CNN (Fuente: Elaboración propia).

Capa	Tamaño kernel o filtro	Número de filtros o kernels.	Tamaño de mapas de características.
Input	-	-	128x128
Convolución 1	5x5	64	124x124
Pooling 1	2x2	64	62x62
Convolución 2	5x5	32	62x62
Pooling 2	2x2	32	31x31
Capa completamente conectada	1x1	512	512
Softmax	-	-	5

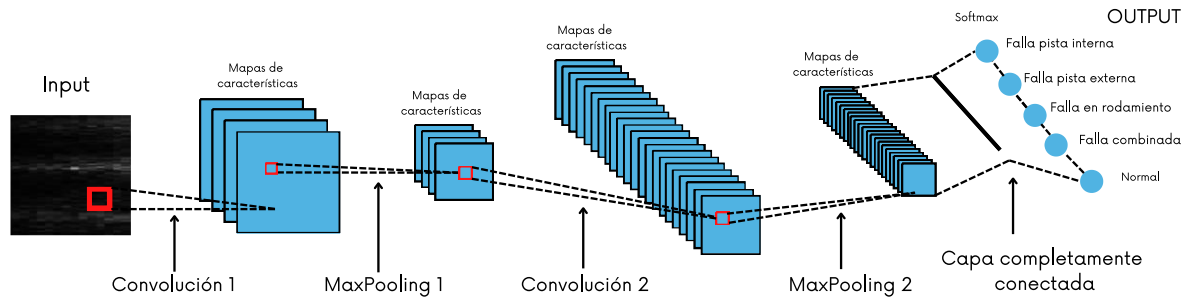


Figura 3.4: Visualización gráfica de la arquitectura de la CNN (Fuente: Elaboración propia.)

La cantidad de parámetros a entrenar por la CNN es de 15,800,997 y la distribución de los datos se describe en la Tabla 3.6. En donde los datos de validación corresponden al 10% de los datos de entrenamiento.

Tabla 3.6: Distribución de datos del banco de imágenes. (Fuente: Elaboración propia.)

Parámetros de entrenamiento	Entrenamiento y validación	Test
Distribución general en porcentajes.	80%	20%

El proceso de entrenamiento para cada banco de imágenes se realiza para 10 épocas y un tamaño de batch de 32 imágenes tiempo-frecuencia. Para obtener resultados generales del modelo, se itera 10 veces el proceso de entrenamiento para cada banco de imágenes. Esto con tal de disminuir la variabilidad producto de la inicialización de pesos de la CNN. En la Figura 3.5, se ilustra el proceso general para el entrenamiento de la CNN.

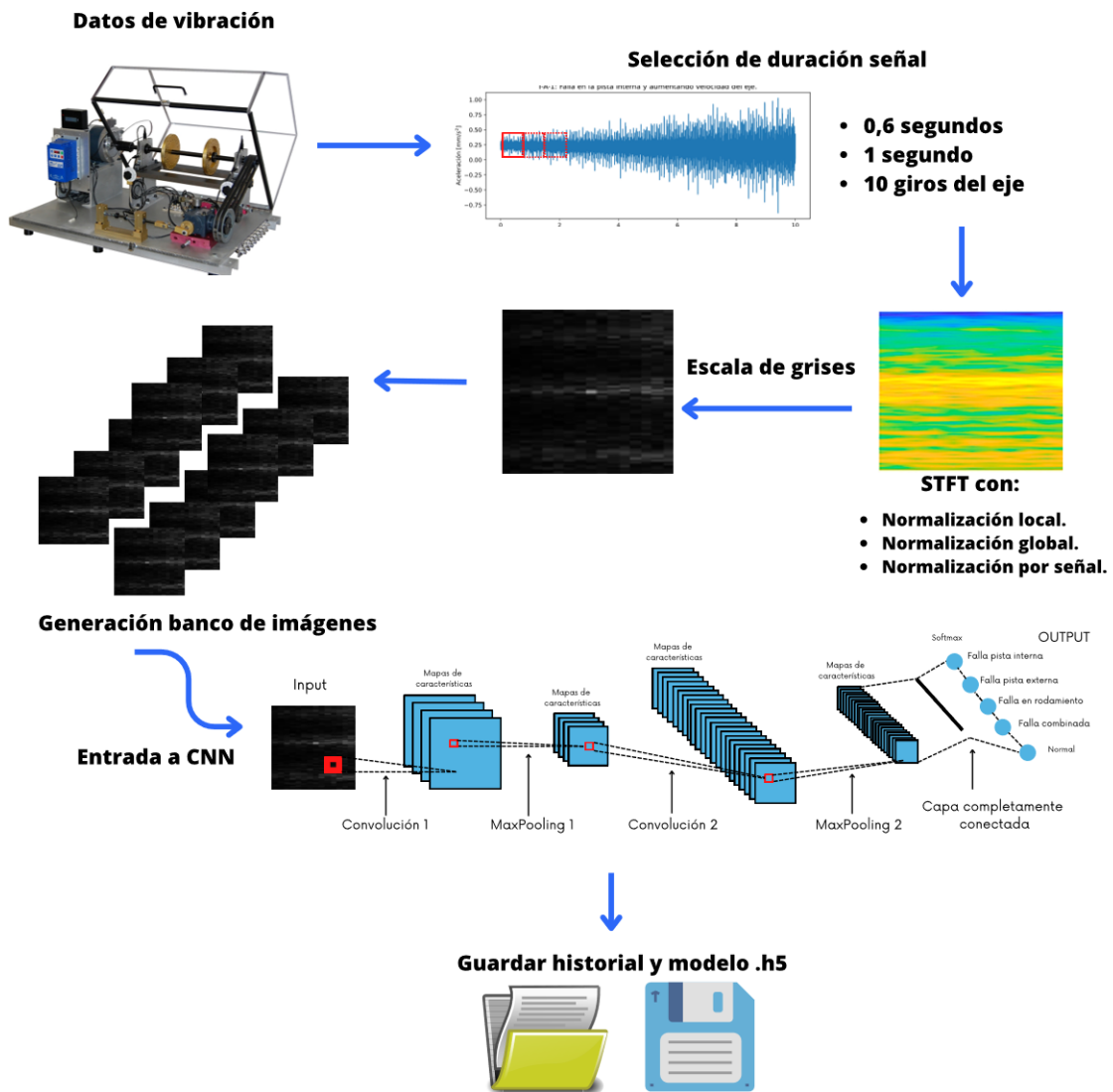


Figura 3.5: Visualización gráfica de la arquitectura de la CNN (Fuente: Elaboración propia.)

En la Tabla 3.7, se detalla la cantidad de modelos .h5, que contienen el registro del entrenamiento y pesos de la CNN. Lo cual se realiza para cada banco de imágenes.

Tabla 3.7: Modelos entrenados de CNN resultantes de la primera etapa. (Fuente: Elaboración propia.).

Duración de señal	Normalización	Cantidad de modelos .h5
0,6 segundos	Global	10
	Local	10
	Por señal	10
1 segundo	Global	10
	Local	10
	Por señal	10
10 giros del eje	Global	10
	Local	10
	Por señal	10
Total	-	90

3.2.3 Selección de la mejor configuración para generar bancos de imágenes mediante STFT.

Una vez obtenida la mejor duración de señal para la generación de los bancos de imágenes, se realiza nuevamente el entrenamiento para generar modelos entrenados de la CNN. Esto se realiza para distintos anchos de ventana y normalizaciones, para la mejor duración de señal seleccionada. En la Tabla 3.8, se describe la cantidad de modelos resultantes para esta etapa.

Tabla 3.8: Modelos entrenados de CNN para distintos anchos de ventana y normalizaciones para mejor duración de señal. (Fuente: Elaboración propia.).

Duración de señal	Ancho de ventana	Cantidad de modelos .h5
Normalización global	4096	10
	8192	10
	16384	10
	32768	10
Normalización local	4096	10
	8192	10
	16384	10
	32768	10
Normalización por señal	4096	10
	8192	10
	16384	10
	32768	10
Total	-	120

Posteriormente del entrenamiento y la selección de la mejor configuración para la generación de las imágenes por la STFT (duración señal, normalización, ancho de ventana). Se procede a seleccionar el mejor modelo .h5 resultante para la configuración seleccionada. Lo que se realiza con tal de utilizar este modelo para evaluar la factibilidad de realizar TL a otro banco de datos.

CAPÍTULO 4: Resultados y discusión.

Este capítulo se estructura en base a los distintos resultados que se obtienen al aplicar la metodología de trabajo anteriormente descrita.

4.1 Duración de señal.

En esta sección se presentan los resultados para la selección de la mejor duración de señal, estableciéndose la comparativa de los resultados de entrenamiento entre los bancos de imágenes de 0,6 s, 1 s y 10 giros del eje. Esta comparativa se realiza considerando un ancho de ventana de 16384 puntos de la señal, 10 épocas y *learning rate* de 0,001.

4.1.1 Resultados de precisión

En las Figuras 4.1, 4.2 y 4.3 se presentan los resultados de precisión promedio para distintas duraciones de señal y normalizaciones. Además, en la Tabla 4.1 se adjuntan detalles adicionales de estos resultados. Los que se obtienen tras compilar 10 veces la CNN, lo que se realiza con tal de descartar la aleatoriedad de los resultados, debido a la inicialización aleatoria de los pesos para el entrenamiento de la CNN.

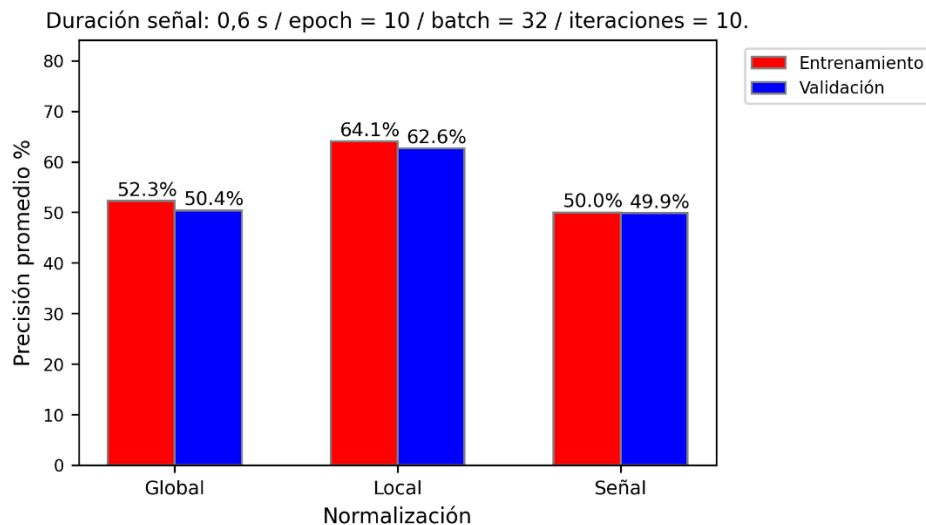


Figura 4.1: Resultados de precisión promedio para duración de señal 0,6 segundos, 10 épocas y *learning rate* 0,001. (Fuente: Elaboración propia.)

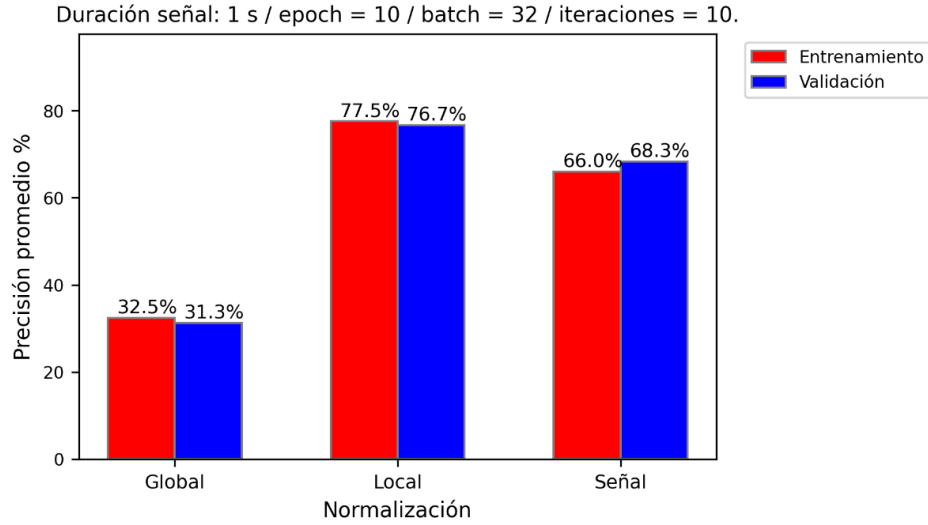


Figura 4.2: Resultados de precisión promedio duración de señal 1 segundo, 10 épocas y *learning rate* 0,001.
(Fuente: Elaboración propia.)

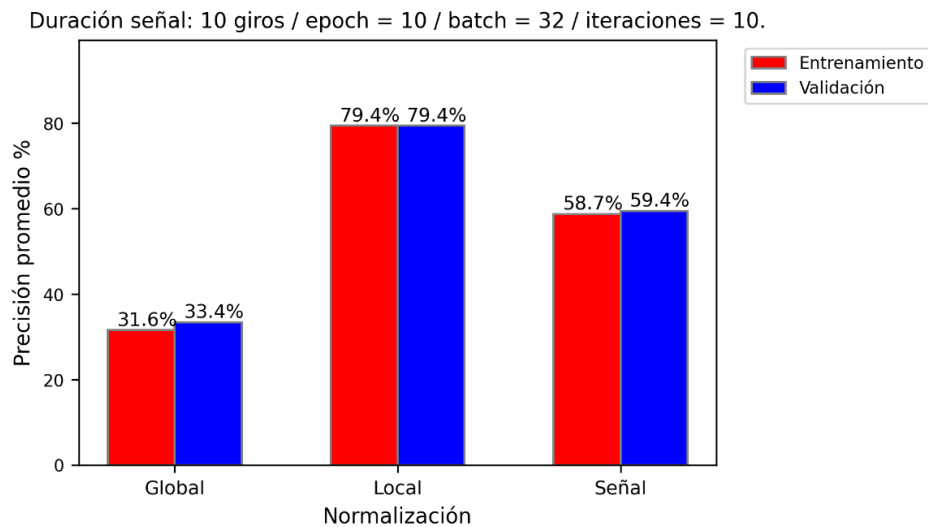


Figura 4.3: Resultados de precisión promedio para duración de señal correspondiente a 10 giros del eje, 10 épocas y *learning rate* 0,001. (Fuente: Elaboración propia.)

Tabla 4.1: Resultados después de 10 iteraciones para cada caso usando datos de validación, 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.).

Duración de señal	Normalización	Máxima precisión	Mínima precisión	Precisión promedio	Desviación estándar
0,6 segundos	Global	86,84%	19,74%	50,39%	25,70
	Local	98,68%	19,74%	62,63%	35,26
	Señal	100,00%	19,74%	49,87%	37,04
1 segundo	Global	66,67%	18,75%	31,25%	19,70
	Local	100%	18,75%	76,67%	30,04
	Señal	97,92%	18,75%	68,33%	32,73
10 giros del eje	Global	88,51%	19,54%	33,45%	25,85
	Local	100%	20,69%	79,43%	29,82
	Señal	100%	20,69%	59,43%	38,75

De los gráficos anteriores, se observa que la precisión promedio presenta los mejores resultados para una normalización local, seguido de la normalización por señal. Siendo la normalización global la que presenta los peores resultados, a simple vista de los gráficos no se puede distinguir claramente la duración de señal que presenta los mejores resultados a lo largo de estas normalizaciones. Por lo que, en la Tabla 4.1 se adjuntan detalles adicionales, en donde se observa que los valores de desviación estándar presentan una gran variabilidad. La que se relaciona directamente con la inicialización de pesos, número de épocas y *learning rate* estipulado para el entrenamiento de la CNN. Los que para este caso son de 10 épocas y *learning rate* de 0,001.

En vista de la variabilidad de los resultados, es necesario realizar un análisis más detallado para evaluar la influencia de los hiperparámetros de entrenamiento de la CNN. Además, de evaluar la influencia de la inicialización de pesos de la CNN, lo que se presenta posteriormente en la sección 4.1.4.

4.1.2 Resultados de precisión y pérdida por épocas.

A continuación, se presentan los gráficos de precisión y pérdida por épocas de los mejores modelos resultantes del entrenamiento de la CNN para esta etapa, que para este caso corresponde a una duración de señal de 10 giros del eje.

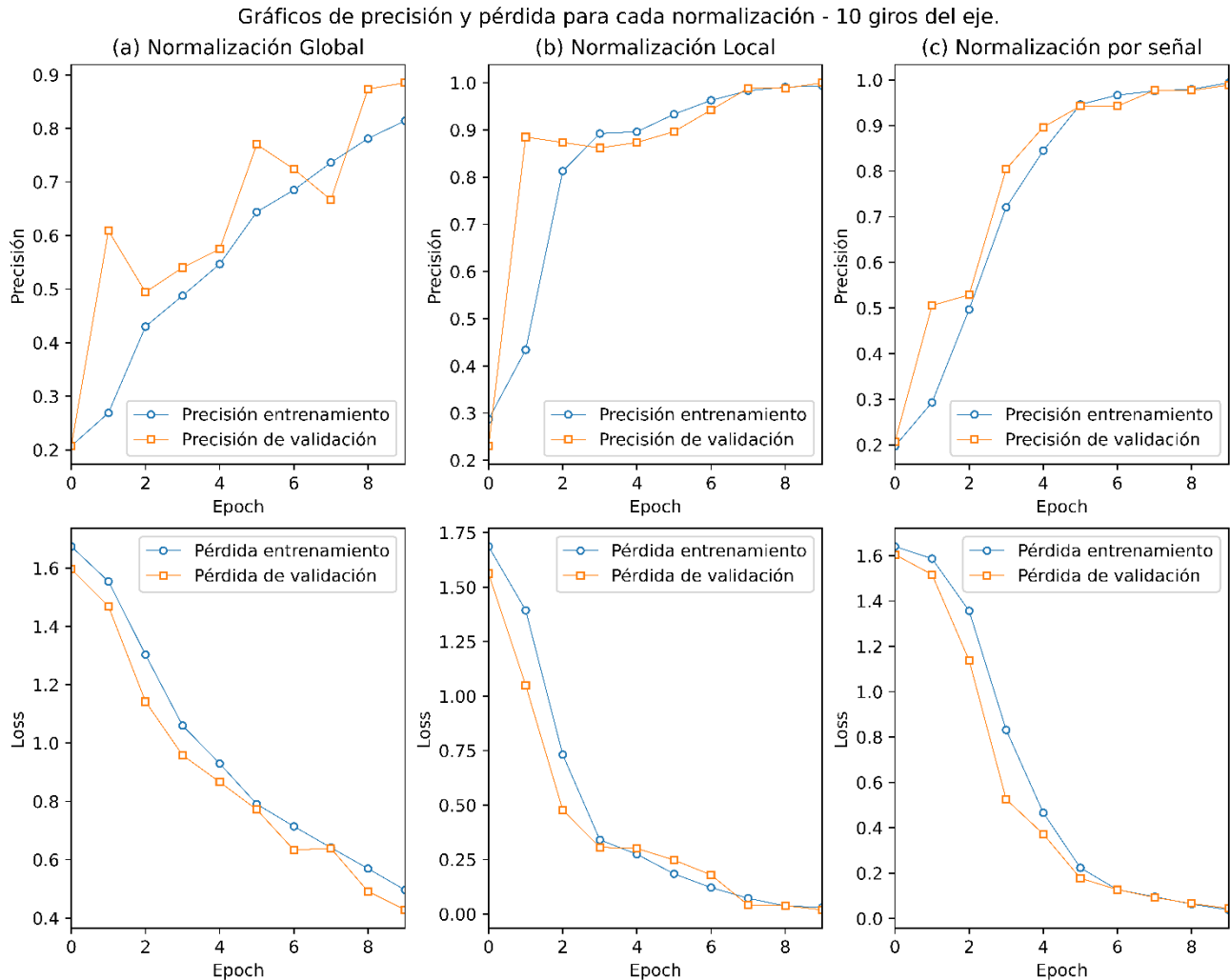


Figura 4.4: Resultados de precisión, pérdida y test para 10 giros del eje (mejores modelos), 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.)

De la Figura 4.4, se tiene que la función de costo converge rápidamente a un valor mínimo en el caso de la normalización local y por señal. Sin embargo, en el caso de la normalización global, el modelo se encuentra con dificultades para extraer características relevantes de la imagen, sin lograr la convergencia. Lo que puede ser producto de la luminosidad de los píxeles.

La luminosidad o intensidad del color se determina con base en la mayor amplitud del amplio espectro de señales a analizar. Por lo que, si una señal contiene una amplitud de vibración dos o más órdenes de magnitud por sobre la señal a procesar, afectaría la calidad de la información que presente la imagen generada y de manera directa el desempeño de la CNN. Esta situación no ocurre en los otros tipos de normalización, dado que se encuentran trabajando en órdenes de magnitud similares para la escala de colores.

4.1.3 Resultados de test para mejores modelos.

Los resultados de los datos de test se presentan en una matriz de confusión en la Figura 4.5. En donde cada matriz corresponde al mejor modelo resultante de la iteración y entrenamiento de la CNN para cada normalización, con una duración de señal de 10 giros del eje.

Matriz de confusión para datos de test - 10 giros por eje.

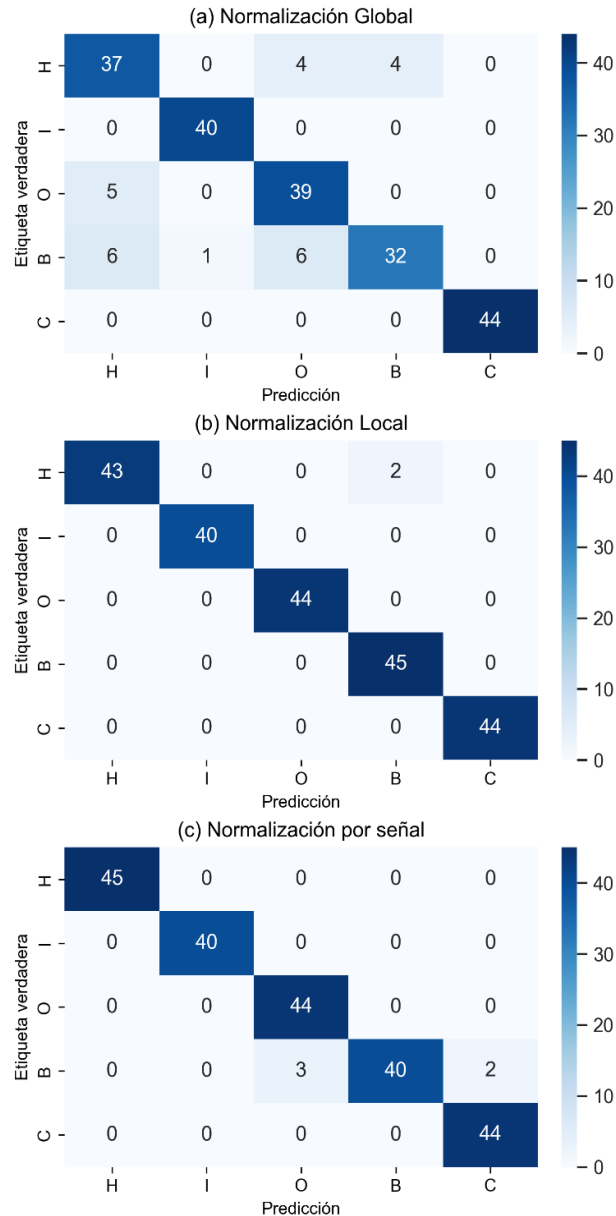


Figura 4.5: Matriz de confusión para datos de test para 10 giros del eje, 10 épocas y *learning rate* de 0,001 (mejores modelos). (Fuente: Elaboración propia.)

Los porcentajes de precisión para los datos de test son de un 88,07%, 99,01% y 97,71% para la normalización global, local y por señal respectivamente. Es de interés señalar que estos porcentajes se deben en menor medida a la inicialización de los pesos realizada en cada iteración de los modelos

entrenados, y en mayor medida a la calidad de la información de la imagen generada e hiperparámetros de entrenamiento de la CNN. Además, no se debe descartar la influencia del tamaño del ancho de ventana, ya que estos resultados se presentan para un tamaño arbitrario de 16384 puntos de la señal. El que puede ser no óptimo para este tipo de configuración y su influencia se estudia en secciones posteriores.

A continuación, se adjuntan algunas métricas adicionales las que se calculan considerando el número de *True Positives*, *False Positives*, *True Negatives* y *False Negatives* para cada clase por separado. Lo que resulta en los valores presentados en la Tabla 4.2, la definición y ejemplo de cálculo de estas métricas se detalla en el Anexo A3. De estas métricas, los mejores valores corresponden a las clases con falla en la pista interna y falla combinada, que coinciden con los archivos que presentan las máximas amplitudes dentro del banco de estudio. Lo que significa que la información que presentan sus imágenes se distingue claramente en cuanto a la luminosidad dada por la escala de colores, comprobándose el cómo afecta el tipo de normalización a usar para la generación de las imágenes T-F.

Tabla 4.2: Reporte de clasificación en datos de test para 10 giros del eje, normalización global, 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.).

Clase	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Condición Normal - H	0,77	0,82	0,80	45
Falla pista interna - I	0,98	1,00	0,99	40
Falla pista externa - O	0,80	0,89	0,84	44
Falla en rodamiento - B	0,89	0,71	0,79	45
Falla combinada - C	1,00	1,00	1,00	44
Total	0,89	0,88	0,88	218

4.1.4 Evaluación influencia hiperparámetros en entrenamiento de la CNN.

A continuación, con tal de evaluar la influencia de los hiperparámetros de entrenamiento de la CNN y descartar la variabilidad de los resultados presentados en la Tabla 4.1. En las Figuras 4.6, 4.7 y 4.8, se presentan los resultados que se obtienen al considerar 30 épocas y un *learning rate* de 0,0001. Similarmente, para verificar la influencia de estos hiperparámetros se adjunta un desglose detallado de los resultados, en la Tabla 4.3.

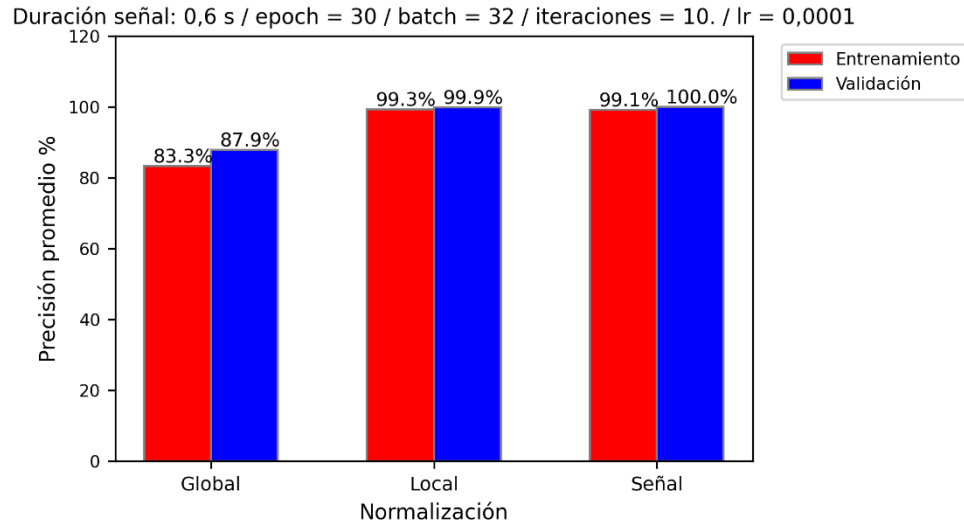


Figura 4.6: Resultados de precisión promedio para duración de señal 0,6 segundos, 30 épocas y *learning rate* 0,0001. (Fuente: Elaboración propia.)

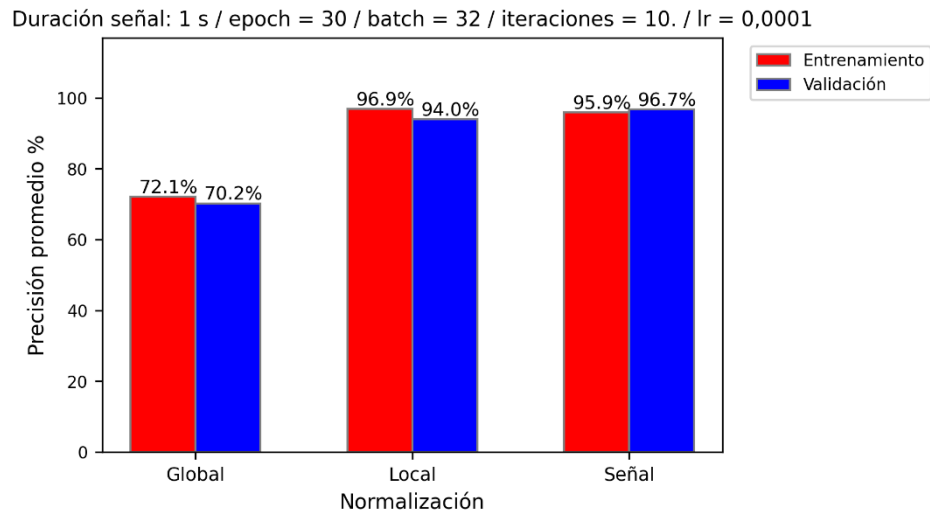


Figura 4.7: Resultados de precisión promedio duración de señal 1 segundo, 30 épocas y *learning rate* de 0,0001. (Fuente: Elaboración propia.)

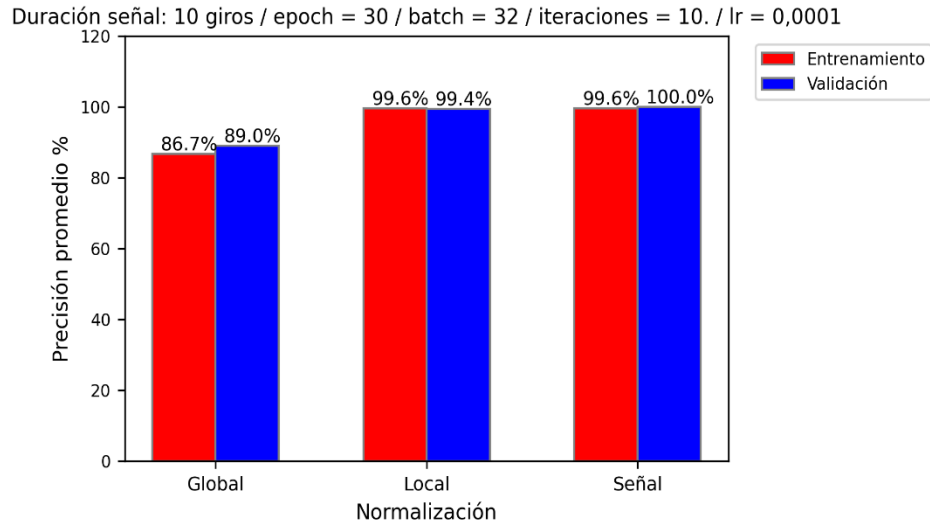


Figura 4.8: Resultados de precisión promedio para duración de señal correspondiente a 10 giros del eje, 30 épocas y *learning rate* de 0,0001. (Fuente: Elaboración propia.)

Tabla 4.3: Resultados después de 10 iteraciones para cada caso usando datos de validación, 30 épocas y *learning rate* de 0,0001. (Fuente: Elaboración propia.).

Duración de señal	Normalización	Máxima precisión	Mínima precisión	Precisión promedio	Desviación estándar
0,6 segundos	Global	89,47 %	85,53 %	87,89 %	1,29
	Local	100,00 %	98,68 %	99,87 %	0,4
	Señal	100,00 %	100,00 %	100,00 %	0,00
1 segundo	Global	79,17 %	58,33 %	70,21 %	8,28
	Local	95,83 %	91,67 %	93,96 %	1,73
	Señal	100,00 %	91,67 %	96,67 %	2,98
10 giros del eje	Global	89,66 %	88,53 %	88,97 %	0,56
	Local	100,00 %	98,68 %	99,87 %	0,77
	Señal	100,00 %	100,00 %	100,00 %	0,00

En los resultados presentados se tiene una baja considerable en cuando a la variabilidad de los resultados de precisión, reflejada en la disminución de la desviación estándar. Lo que implica que el *learning rate* y número de épocas tienen gran influencia en el entrenamiento de la CNN. Además, se observa que la influencia de la inicialización de pesos se ve disminuida, dado que la mínima precisión obtenida en los 10 modelos es de un 85% en adelante, lo que se presenta para la gran mayoría de configuraciones, a excepción de la duración de señal de 1 s y normalización global. Lo que significa un aumento considerable en la precisión mínima, en comparación a los resultados presentados en la Tabla 4.1.

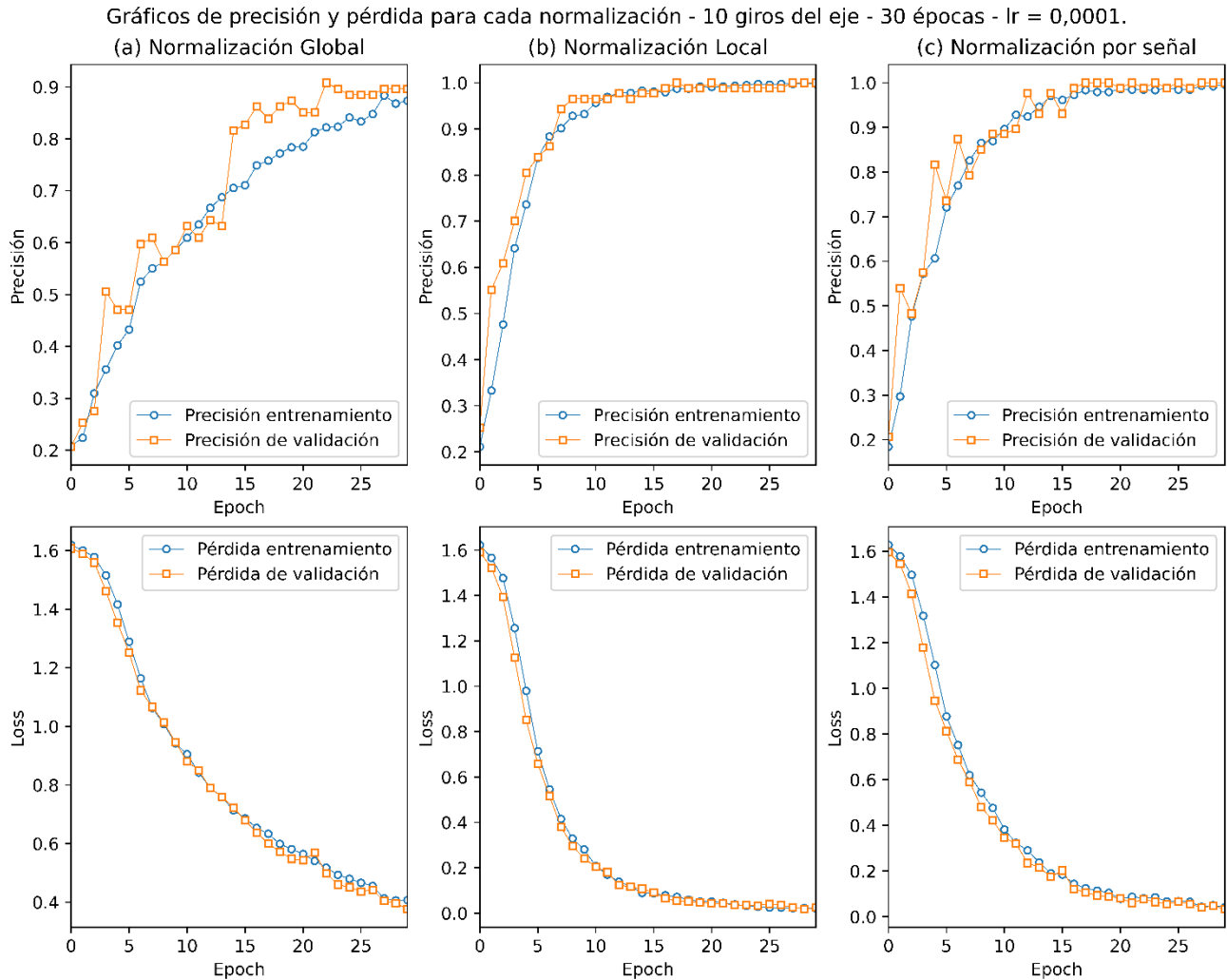


Figura 4.9: Resultados de precisión, pérdida y test para 10 giros del eje (mejores modelos), 30 épocas y $learning$ rate de 0,0001.

En la Figura 4.9, se observa que el $learning$ rate y número de épocas son hiperparámetros importantes a considerar en el entrenamiento de la CNN. Puesto que el $learning$ rate, determina la magnitud de ajuste de los pesos durante cada época, a través de la función de costo. Lo que se traduce en que un mayor $learning$ rate provoque una convergencia más rápida, pero se corre el riesgo de que el modelo no converja en absoluto. Por lo que, al optar por un $learning$ rate más pequeño, se logra disminuir la variabilidad de los resultados y alcanzar la convergencia. Por otra parte, un mayor número de épocas significa que el modelo ajuste los pesos y mejore su rendimiento. Sin embargo, también se debe tener cuidado con este hiperparámetro, puesto que puede provocar que el modelo se ajuste demasiado a los datos de entrenamiento. Lo que se denomina sobre ajuste u *overfitting*, disminuyendo la capacidad de generalización del modelo con los datos de validación o test.

La duración de señal influye directamente en la cantidad de datos disponibles que se pueden generar. Lo que se refleja en los resultados obtenidos para una duración de señal de 0,6 s y 10 giros del eje, que generan una cantidad de datos disponibles similares y poseen los mayores porcentajes de precisión promedio. En donde se tienen 944 y 1086 datos disponibles para una duración de 0,6 s y 10 giros del

eje respectivamente. En cuanto a la duración de señal de 1 segundo, es tienen solamente 590 datos disponibles para el entrenamiento, provocando que esta duración tenga los peores resultados entre las duraciones de señal evaluadas. En resumen, mayor cantidad de datos disponibles significa un mejor desempeño de la CNN, mientras que una baja disponibilidad implica un deterioro de este desempeño.

Por otra parte, es necesario tener en consideración las características no-estacionarias de la señal que afectan directamente los casos de una duración de 0,6 s y de 1 segundo del eje. En donde se tiene como resultado que las distintas imágenes generadas contengan variaciones en las características presentadas entre unas y otras, dada las variaciones en la velocidad del eje. Lo que pone a prueba la capacidad de la CNN para detectar estos cambios en las imágenes, lo que supone el que esta sea capaz de reconocer características representativas de los modos de falla, independientemente de su posición en la imagen.

De esta etapa se selecciona una duración de señal de 10 giros del eje para evaluar la influencia de las demás configuraciones. Sin embargo, la influencia de esta configuración se debe reevaluar, considerándose distintos anchos de ventana para una normalización local, lo que se realiza en secciones posteriores.

4.2 Resultados para distintos anchos de ventana y normalizaciones.

La evaluación de las configuraciones de normalización y ancho de ventana se realiza para el banco de imágenes seleccionado, con una duración de señal de 10 giros del eje. Los resultados de esta evaluación se presentan en las Figuras 4.10, 4.11 y 4.12.

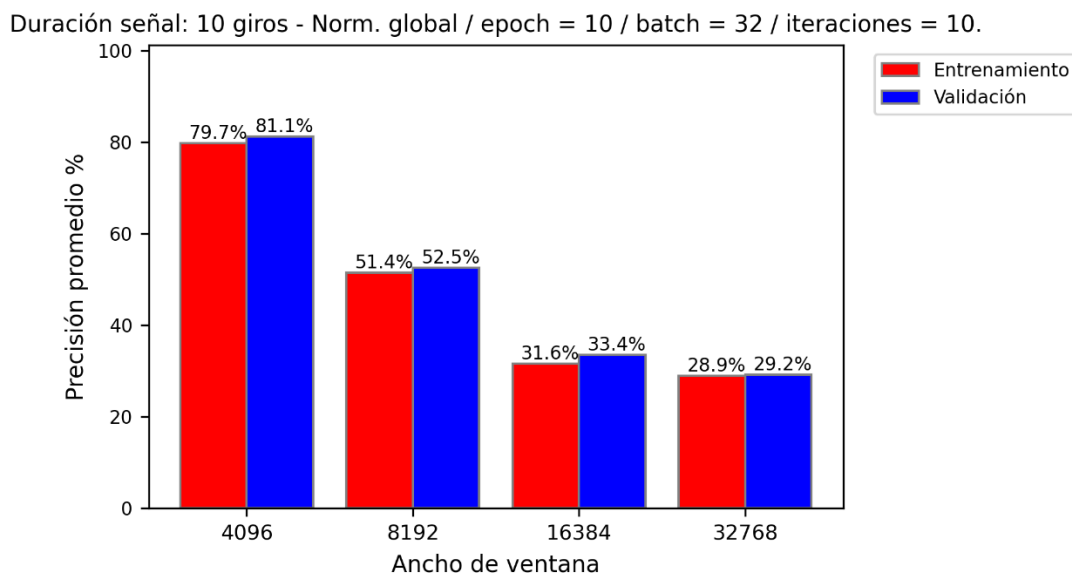


Figura 4.10: Resultados de precisión promedio para duración 10 giros del eje, normalización global, 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.)

Duración señal: 10 giros - Norm. local / epoch = 10 / batch = 32 / iteraciones = 10.

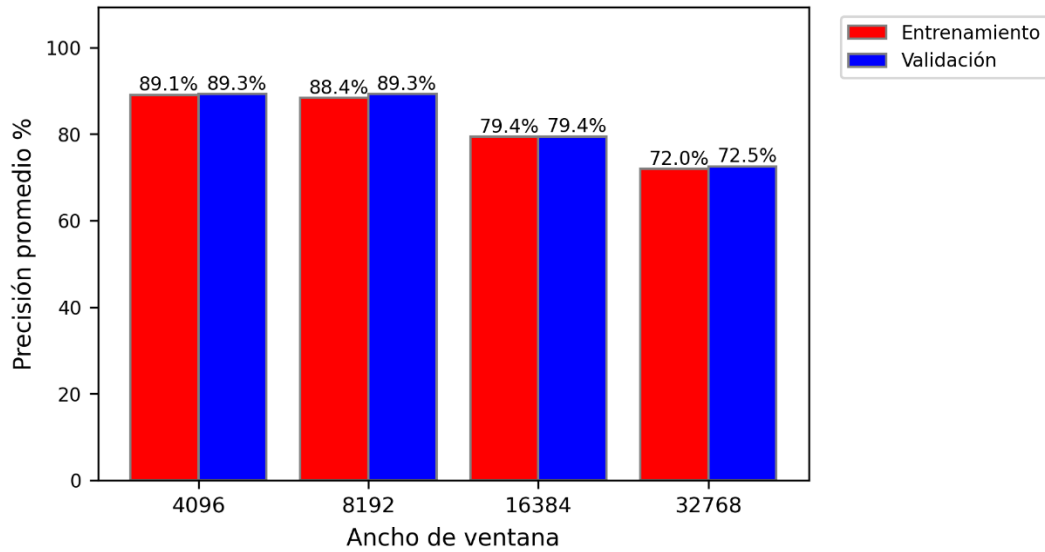


Figura 4.11: Resultados de precisión promedio para duración 10 giros del eje, normalización local, 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.)

Duración señal: 10 giros - Norm. por señal / epoch = 10 / batch = 32 / iteraciones = 10.

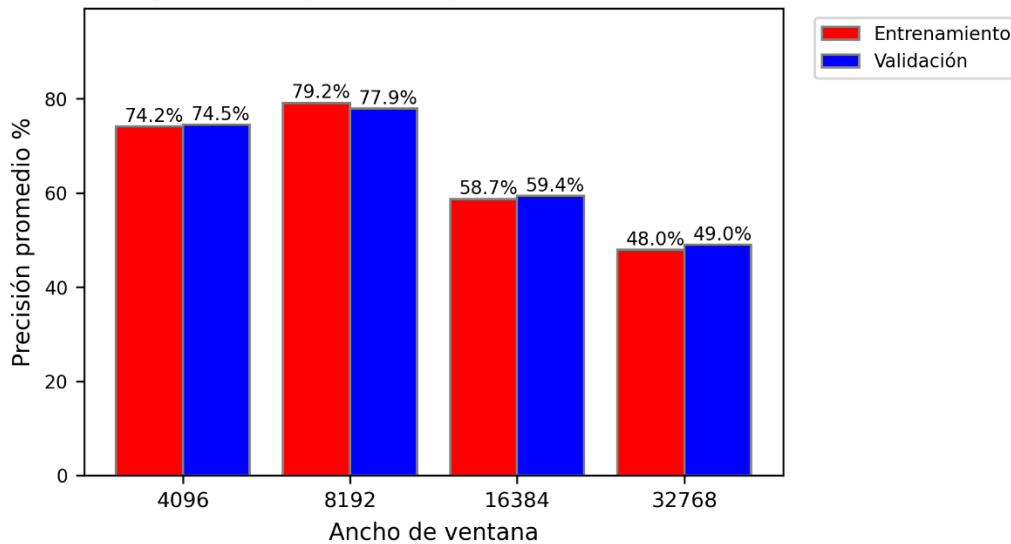


Figura 4.12: Resultados de precisión promedio para duración 10 giros del eje, normalización por señal, 10 épocas y *learning rate* de 0,001. (Fuente: Elaboración propia.)

De los gráficos en la Figura 4.11, se visualiza que el mejor valor de precisión promedio se alcanza para una normalización local y un ancho de ventana de 4096 puntos de la señal. Además, se visualiza cierta tendencia en un deterioro de la precisión promedio a medida que aumenta el ancho de ventana.

A continuación, se presentan los resultados y métricas del mejor modelo para este ancho de ventana. En los que se consideran una configuración de normalización local, duración de señal de 10 giros del eje y ancho de ventana de 4096 puntos de la señal.

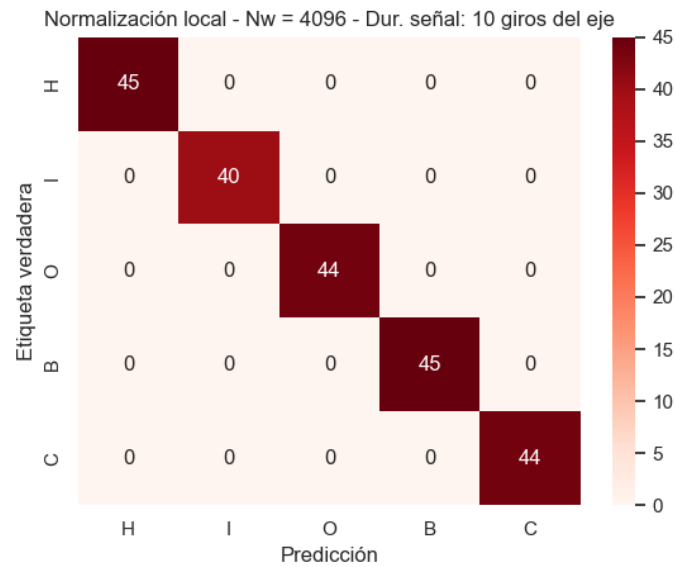


Figura 4.13: Resultados de datos de test normalización local, 10 giros del eje, ancho de ventana 4096 puntos de la señal. (Fuente: Elaboración propia.)

De la Figura 4.13, se observa que el modelo es capaz de clasificar correctamente las 5 clases o modos de fallas presentes en la imagen con un 100% de precisión. En la Tabla 4.4, se resumen otras métricas que reflejan el desempeño del mejor modelo resultante de las 10 iteraciones, para las configuraciones mencionadas.

Tabla 4.4: Reporte de clasificación en datos de test para mejores configuraciones. (Fuente: Elaboración propia.)

Clase	Precision	Recall	F1-Score	Support
Condición Normal - H	1,00	1,00	1,00	45
Falla pista interna - I	1,00	1,00	1,00	40
Falla pista externa - O	1,00	1,00	1,00	44
Falla en rodamiento - B	1,00	1,00	1,00	45
Falla combinada - C	1,00	1,00	1,00	44
Total	1,00	1,00	1,00	218

En cuanto a los resultados obtenidos para esta sección, se tiene que la influencia del tipo de normalización afecta en la capacidad de representar o resaltar las características de las imágenes generadas por la STFT. Lo que repercute directamente en el desempeño de la CNN y su capacidad de distinguir entre una clase y otra.

El tamaño del ancho de ventana corresponde a una configuración importante a considerar, dado que los mejores resultados se presentan al utilizar un ancho de ventana más pequeño. Este ancho de ventana aumenta la resolución en el dominio tiempo, lo que permite visualizar cambios repentinos en la señal de vibración, como puede ser los impactos asociados a la pista interna o externa, por ejemplo. Esta configuración es de tal influencia que facilita la capacidad de la CNN para reconocer

características relevantes de cada imagen a pesar de las variaciones temporales producidas por la señal no-estacionaria.

Por otra parte, el ancho de ventana para el caso de la normalización global presenta grandes variaciones entre distintos anchos. Por lo que para este tipo de normalización se debe seleccionar de manera adecuada el ancho de ventana, con tal de resaltar en detalle las características de los modos de falla y que la CNN sea capaz de entrenarse correctamente. Este detalle en la imagen depende directamente de la escala utilizada para normalizar la escala de colores. Finalmente, se concluye que la normalización global no es recomendable para este caso de estudio dada la variabilidad de sus resultados.

4.3 Reevaluación de la duración de señal y anchos de ventanas.

Para esclarecer las configuraciones anteriormente seleccionadas, a continuación, se presentan los resultados para distintos anchos de ventana y duraciones de señal. Lo que se realiza considerando una normalización local, puesto que es la que entrega los mejores resultados de precisión máxima y promedio presentes en la Tabla 4.1 y Tabla 4.3.

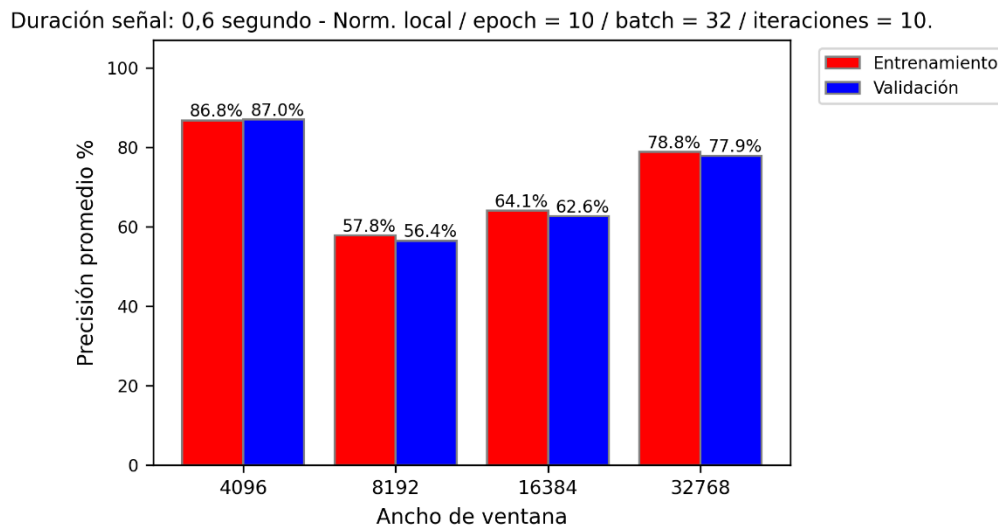


Figura 4.14: Resultados de precisión promedio para duración 0,6 segundo y normalización local. (Fuente: Elaboración propia.)

Duración señal: 1 segundo - Norm. local / epoch = 10 / batch = 32 / iteraciones = 10.

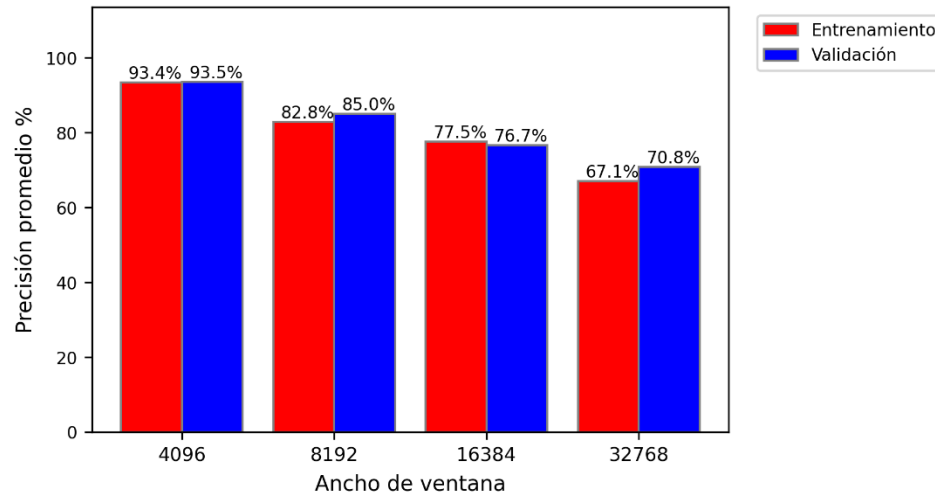


Figura 4.15: Resultados de precisión promedio para duración 1 segundo y normalización local. (Fuente: Elaboración propia.)

Duración señal: 10 giros - Norm. local / epoch = 10 / batch = 32 / iteraciones = 10.

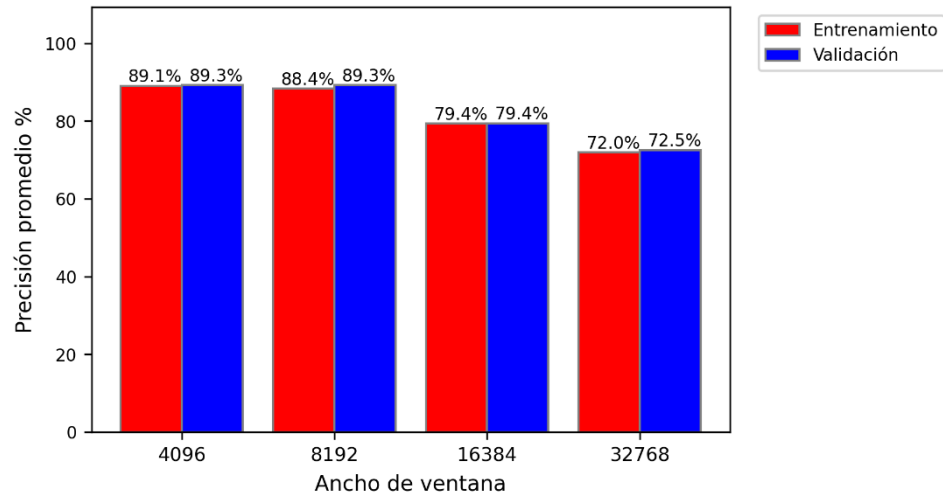


Figura 4.16: Resultados de precisión promedio para duración 10 giros del eje y normalización por local. (Fuente: Elaboración propia.)

De las Figuras 4.14, 4.15 y 4.16 se tiene que la duración de señal que presenta los mejores resultados en cuanto a precisión promedio para los distintos anchos de ventana, es la duración de señal de 10 giros del eje. Para la duración de señal de 1 segundo se observa una tendencia decreciente en la precisión promedio, a medida que se aumenta el ancho de ventana. Esta duración presenta resultados parecidos a los de una duración de 10 giros del eje, a pesar de solo contar con 590 imágenes para el entrenamiento desde cero de la CNN. Sin embargo, se tiene que esta señal posee la mayor duración entre las presentadas. Lo que significa que para un ancho de ventana de 4096 esta sea capaz de generar imágenes con mayor cantidad de información T-F, asociada a los modos de falla.

En cuanto a duración de señal de 0,6 segundos, se observa cierta variabilidad y la ausencia de una tendencia decreciente asociada al aumento del ancho de ventana. Lo que se debe principalmente a que las imágenes generadas para este banco no dispongan de información relevante asociada a los cambios temporales en la señal, producto de las variaciones en la velocidad del eje. Lo que dificulta la capacidad de la CNN para aprender características relevantes.

En resumen, las mejores configuraciones corresponden a una duración de señal de 10 giros del eje, normalización local y ancho de ventana pequeño de 4096 puntos de la señal.

4.4 Visualización de características aprendidas mediante t-SNE.

En las Figuras 4.17, 4.18 y 4.19, se muestran la distribución de las características aprendidas por el modelo para una duración de señal de 10 giros del eje, ancho de ventana de 4096 puntos y distintas normalizaciones. Estos modelos corresponden a los que tienen los mejores resultados de las 10 iteraciones realizadas.

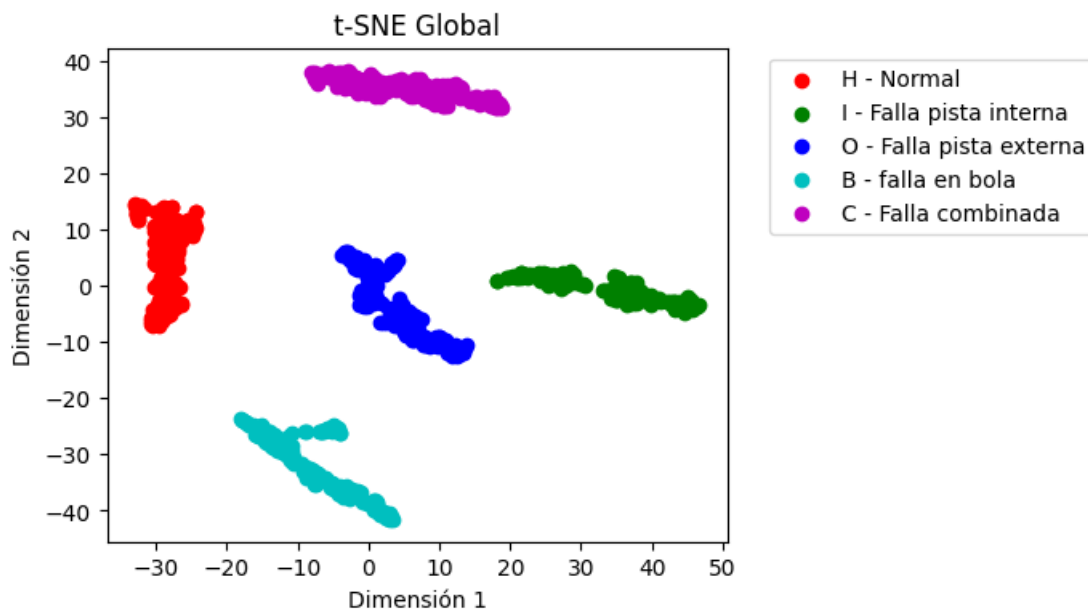


Figura 4.17: Resultados t-SNE para 10 giros, normalización global y 4096 ancho de ventana. (Fuente: Elaboración propia.)

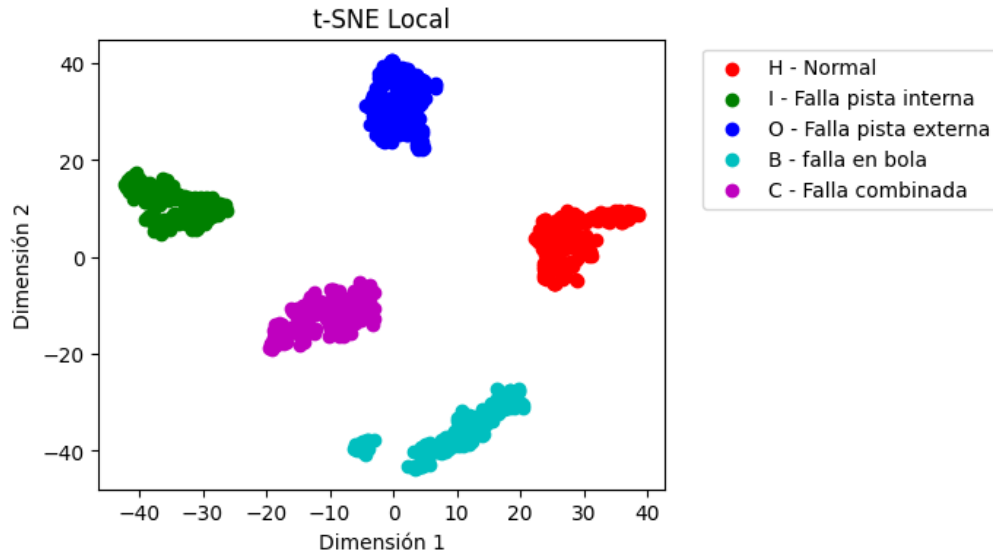


Figura 4.18: Resultados t-SNE para 10 giros, normalización local y 4096 ancho de ventana. (Fuente: Elaboración propia.)

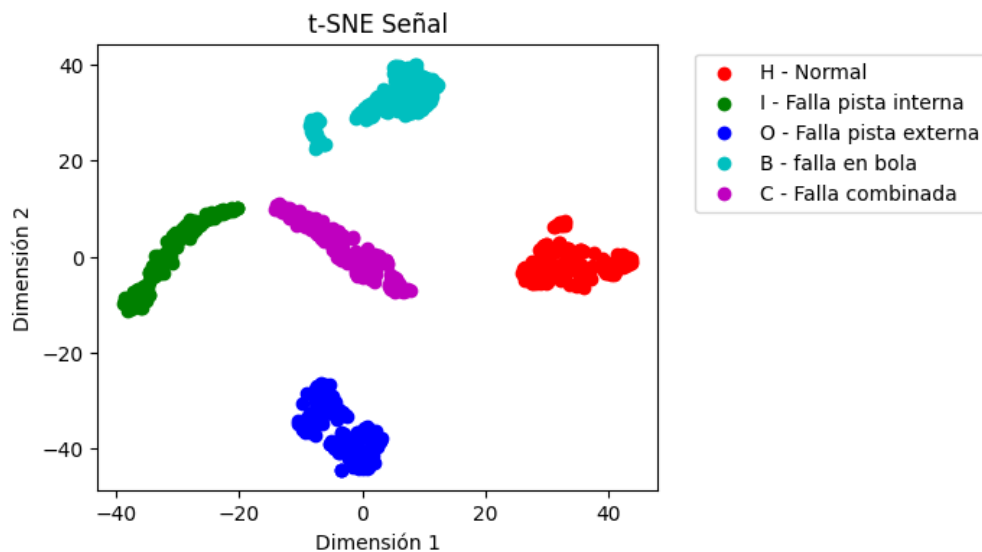


Figura 4.19: Resultados t-SNE para 10 giros, normalización por señal y 4096 ancho de ventana. (Fuente: Elaboración propia.)

Los resultados de estas imágenes se obtienen al aplicar el método *t-distributed Stochastic Neighbor Embedding* (t-SNE). El t-SNE corresponde a un método de reducción de dimensionalidad, el cual se usa para mapear datos de altas dimensiones a un espacio de menor dimensión. De manera que, aplicándose este método a las características aprendidas por la CNN, se puede evaluar la calidad o eficacia con que este modelo aprendió las características de los datos [22].

De los gráficos de las Figuras 4.17, 4.18 y 4.19, se tiene que la CNN es capaz de agrupar correctamente las clases para las distintas normalizaciones. Lo que refleja la robustez y capacidad de generalización

del modelo resultante de la selección adecuada de las configuraciones estudiadas. En donde la configuración que presenta la mayor influencia es el ancho de ventana a considerar, y que se debe seleccionar en relación con el fenómeno a estudiar. Es de tal importancia esta configuración, que del gráfico de la Figura 4.17, se observa que para la normalización global se logra distinguir claramente todas las clases. Mitigándose el efecto que se encuentra asociado a este tipo de normalización, como lo es la luminosidad de los píxeles.

En resumen, se tiene que los resultados obtenidos por la t-SNE son congruentes a los presentados en la Sección 4.3, siendo el mejor modelo resultante del banco de datos con una normalización local, 10 giros del eje y ancho de ventana de 4096 puntos de la señal.

4.5 Visualización de mapas de características para distintos tipos de falla.

A continuación, se presentan los mapas de características que se obtienen al aplicar los filtros aprendidos durante el entrenamiento de la CNN. Lo que se realiza para imágenes que presentan distintas condiciones en su espectrograma.



Figura 4.20: Mapas de características para falla en bola, imagen: B-A-1_0. (Fuente: Elaboración propia.)

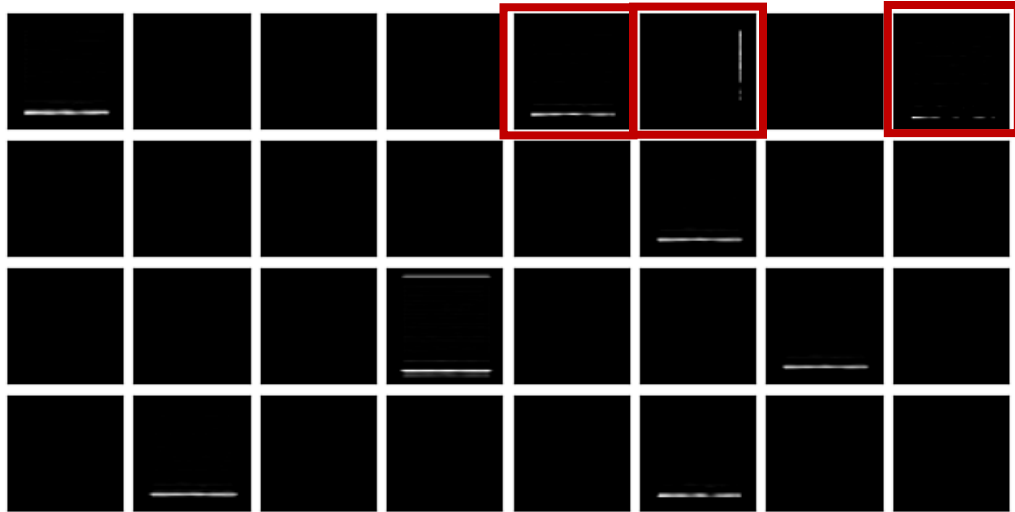


Figura 4.21: Mapas de características para falla combinada, imagen: C-A-1_0. (Fuente: Elaboración propia.)

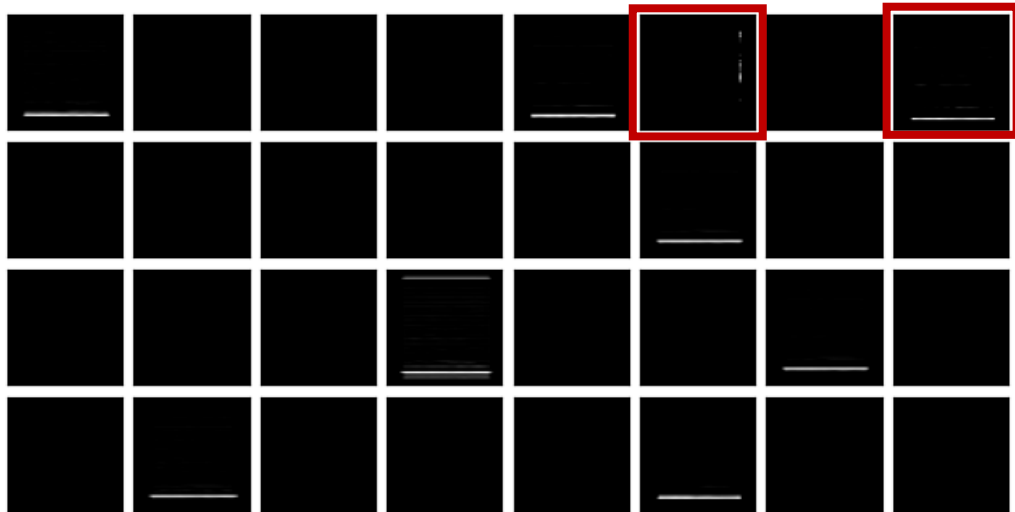


Figura 4.22: Mapas de características para falla pista externa, imagen: O-A-1_0. (Fuente: Elaboración propia.)

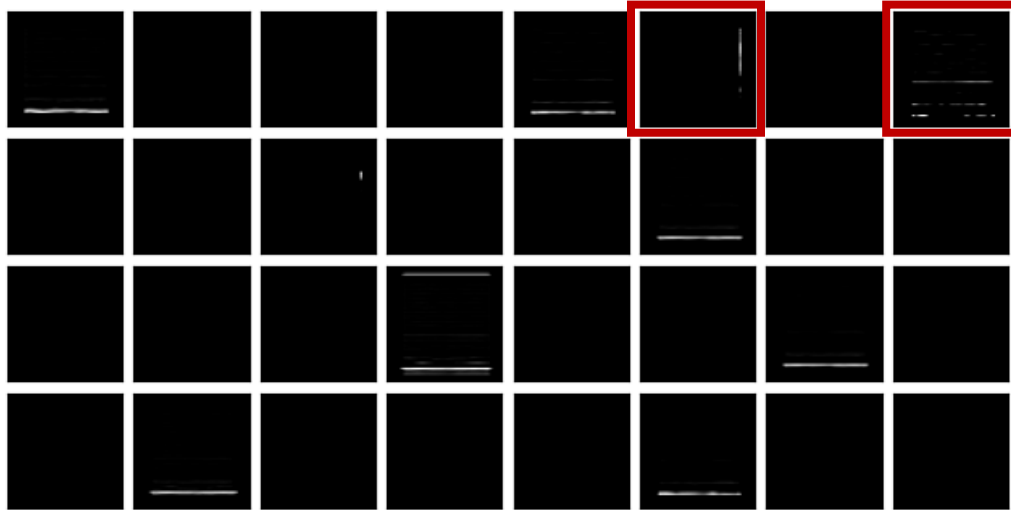


Figura 4.23: Mapas de características para falla pista interna, imagen: I-A-2_0. (Fuente: Elaboración propia.)

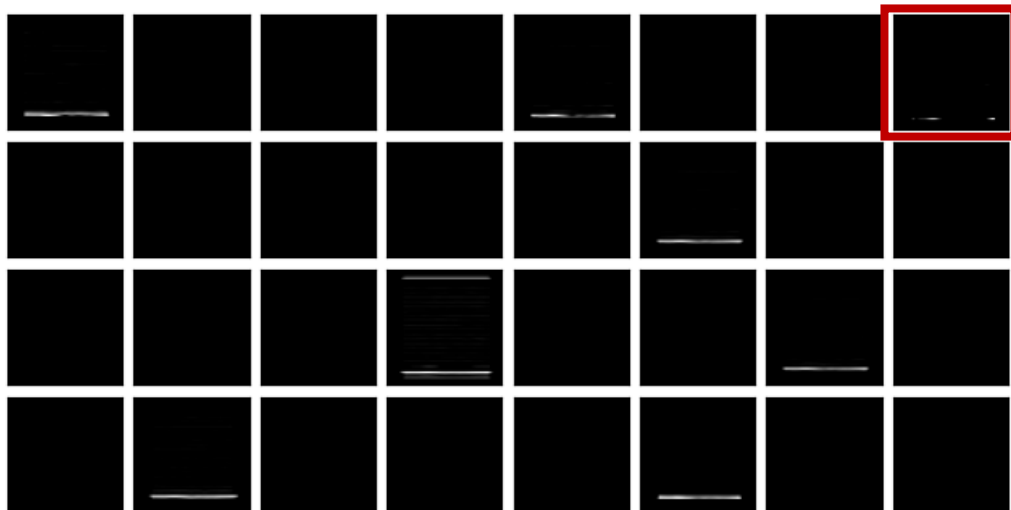


Figura 4.24: Mapas de características para condición normal, imagen: H-A-1_0. (Fuente: Elaboración propia.)

De las Figuras 4.20, 4.21, 4.22, 4.23 y 4.24 se observa que la aplicación de estos filtros genera respuestas distintivas en los mapas de características. Esta activación distintiva se señala encerrada por un rectángulo en rojo, la que varía en cuanto a la intensidad y aparición de distintas líneas en los mapas de características. La que es la forma que tiene la CNN para representar cada falla en particular de este caso de estudio.

Los mapas de características de color negro son mapas de características inactivos y los mapas similares entre sí representan información redundante. Lo que significa que el proceso para la generación de los bancos de imágenes se puede seguir refinando [22]. De tal manera que estas imágenes contengan más información con respecto al tipo de falla que representan, lo que en consecuencia significaría una mayor activación de los mapas de características. Además, se debe considerar que estos mapas de características corresponden a representaciones más abstractas de las imágenes dada su profundidad en la arquitectura de la CNN. En donde las respuestas de estos mapas se ubican previos a la entrada a la capa completamente conectada.

4.6 Comparación con modelo de regresión logística.

Para evaluar que el modelo entrenado no se encuentre sobredimensionado para la tarea de clasificación a realizar. Se realiza la comparación con un modelo de regresión logística, en donde el entrenamiento se realiza con respecto a los atributos: promedio de píxeles de cada imagen y desviación estándar de los píxeles de cada imagen.

Este entrenamiento se realiza con una normalización y búsqueda de malla de los mejores hiper-parámetros para este modelo. Los hiper-parámetros resultantes de la búsqueda de malla son: regularización de $C=1000$ y solucionador ‘newton-cg’.

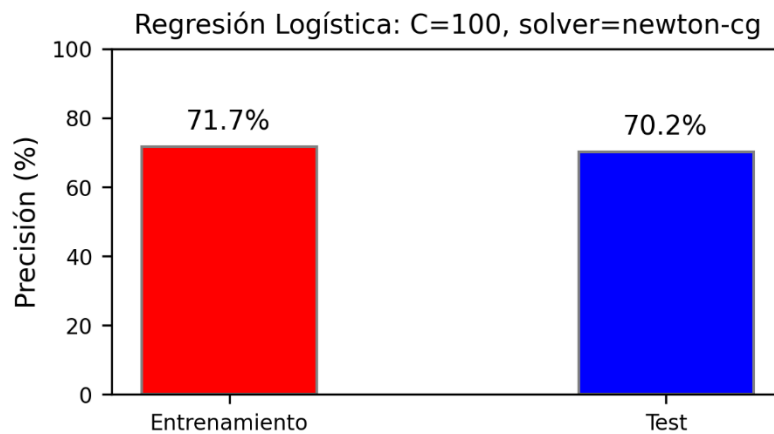


Figura 4.25: Resultados de la regresión logística. (Fuente: Elaboración propia.)

Del gráfico de la Figura 4.25, los porcentajes obtenidos por este método son bastante bajos, alcanzando un 70,2% para los resultados de test. Por lo que se justifica el uso de un modelo más complejo, como lo es la CNN la cual alcanza resultados del 100% para los datos de test en esta tarea multiclase.

CAPÍTULO 5: *Transfer Learning*.

En este capítulo se evalúa la factibilidad de realizar *Transfer Learning* (TL) a un nuevo banco de datos.

5.1 Contexto.

La efectividad de los modelos de ML y DL depende en gran manera de la cantidad de datos disponibles y etiquetados para su entrenamiento. Sin embargo, en la mayoría de las aplicaciones de la industria esto no es factible, producto principalmente de las siguientes razones [23]:

1. La baja cantidad de datos de fallas disponibles, producto del riesgo asociado a que la máquina se encuentre operando con una falla. Además, de que la mayoría del tiempo la máquina se encuentra operando en condiciones normales.
2. La variedad de distintas condiciones de operación, cargas y velocidades. Lo que implica tener una mayor cantidad de datos recopilados para cubrir las posibles condiciones de funcionamiento.
3. La necesidad de monitorear durante un periodo de tiempo prolongado la máquina, con tal de cubrir los procesos de degradación previos a la ocurrencia de una falla.

En respuesta a estas problemáticas, se propone el uso de la TL. Los métodos de TL se basan en utilizar la similitud entre datos de distintos dominios y aplicar el conocimiento aprendido en un dominio fuente a un dominio objetivo. El dominio fuente se refiere al dominio que cuenta con el conocimiento y una gran cantidad de datos disponibles correctamente etiquetados. Por otra parte, el dominio objetivo hace referencia a aquel dominio del cual se cuenta con pocos datos disponibles y etiquetados [24]. En la Figura 5.1, se presenta una analogía entre distintos dominios representando la idea del TL.

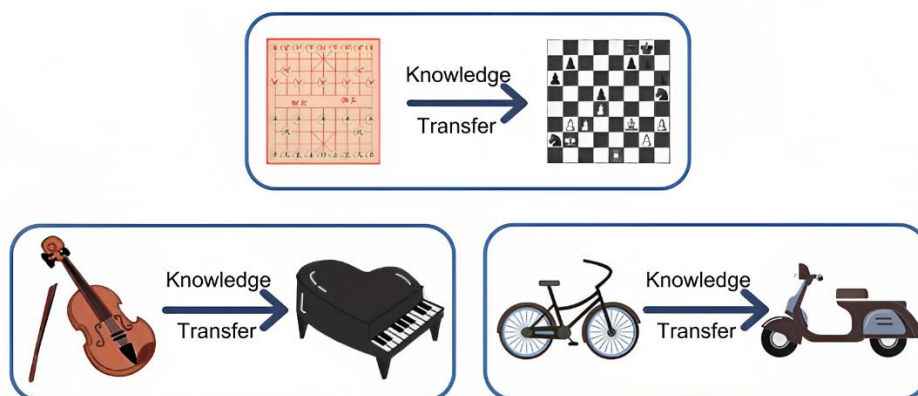


Figura 5.1: Ejemplos intuitivos de la Transferencia de conocimiento. (Fuente: [25])

Entre los métodos de TL más utilizados se encuentran: TL basado en instancias, modelos y características [26]. El TL basado en modelos se basa en compartir la estructura de la red neuronal y

los parámetros entre el dominio fuente y objetivo. Este se utilizará para evaluar la factibilidad de la aplicación de TL a un nuevo caso de estudio.

5.2 Banco de datos objetivo y metodología.

El banco de datos a utilizar para evaluar la factibilidad del TL corresponde al publicado por la *American Society for Machinery Fault Prevention Technology* [27]. Este banco se compone de datos experimentales y reales, de los cuales se seleccionan los datos experimentales como caso de estudio.

Los datos experimentales se componen de datos con condiciones normales, falla en la pista interna y externa para distintas cargas y una velocidad del eje constante de 25 Hz. El detalle de la estructura de los datos se presenta en la Tabla 5.1.

Tabla 5.1: Estructura banco de datos y casos de estudio. (Fuente: [27]).

Clase	Carga [lb]	Frecuencia de muestreo [Hz]	Duración [s]	Número de puntos de datos.
Normal	270	97 756	6	585 936
Falla en pista externa	25	48 828	3	146 384
	50	48 828	3	146 384
	100	48 828	3	146 384
	150	48 828	3	146 384
	200	48 828	3	146 384
	250	48 828	3	146 384
	300	48 828	3	146 384
Falla en pista interna	0	48 828	3	146 384
	50	48 828	3	146 384
	100	48 828	3	146 384
	150	48 828	3	146 384
	200	48 828	3	146 384
	250	48 828	3	146 384
	300	48 828	3	146 384

Las características de los rodamientos para un ángulo de contacto de 0° se presentan en la Tabla 5.2.

Tabla 5.2: Parámetros de los rodamientos de banco de datos MFPT. (Fuente: [27]).

Modelo rodamiento	Diámetro Pitch (mm)	Diámetro bolas (mm)	Número de bolas	BPMI (Hz)	BPFO (Hz)
NICE Bearing	31.62	5.96	8	118.87	81.12

5.3 Generación de banco de imágenes.

Similarmente a la metodología planteada en la Sección 3.2.1, se procede a generar los bancos de imágenes para el caso de estudio. Los bancos se generan utilizando las mejores configuraciones de duración de señal, normalización y anchos de ventana obtenidos en el primer banco de datos presentado en esta memoria.

La estructura de los bancos de imágenes se diseña de tal manera de utilizar una cantidad de giros del eje fija, generándose cierta cantidad de imágenes disponibles para el entrenamiento. El ancho de ventana seleccionado es de 1024 puntos, lo cual es realizado de manera arbitraria según los resultados obtenidos anteriormente y con tal de que contenga la mayor cantidad de información relativa al modo de falla presente en la señal. El detalle de las imágenes generadas para los bancos se presenta en la Tabla 5.3.

Tabla 5.3: Estructura de los bancos de imágenes MFPT. (Fuente: Elaboración propia.).

Duración de señal	Normalización	Número de modos de falla.	Cantidad total de imágenes generadas
3 giros del eje	Local	3	500
10 giros del eje	Local	3	143

La distribución de los datos de test y falla varía para según la siguiente proporción, 30% para datos de entrenamiento y 70% para datos de test para cada banco de datos con tal de evaluar la capacidad de generalización del modelo.

5.3.1 Metodología para la aplicación de TL.

La metodología para la aplicación de TL en este nuevo banco de datos se basa en el *fine-tuning*, el cual toma en cuenta el conocimiento adquirido en un modelo pre-entrenado. El modelo pre-entrenado en este caso corresponde al obtenido en la primera parte de resultados para una normalización local, 10 giros del eje y un ancho de ventana de 16384 puntos de la señal.

En el *fine-tuning*, los pesos aprendidos en las capas inferiores del modelo pre-entrenado se comparten con los del modelo objetivo, por lo que solo se entrenan las capas superiores del modelo. Esto se logra al congelar los pesos de las capas inferiores y actualizar solamente los pesos de las capas superiores [24]. La eficacia de este método es influenciada por la similitud entre el dominio fuente y objetivo. Lo que implica que, para dominios con una alta similitud entre sí solamente se deba ajustar los pesos de la capa completamente conectada. En cambio, para dominios con baja similitud puede ser necesario añadir capas adyacentes o adicionales al modelo objetivo. En la Figura 5.2, se presenta un esquema representando el concepto de *fine-tuning*.

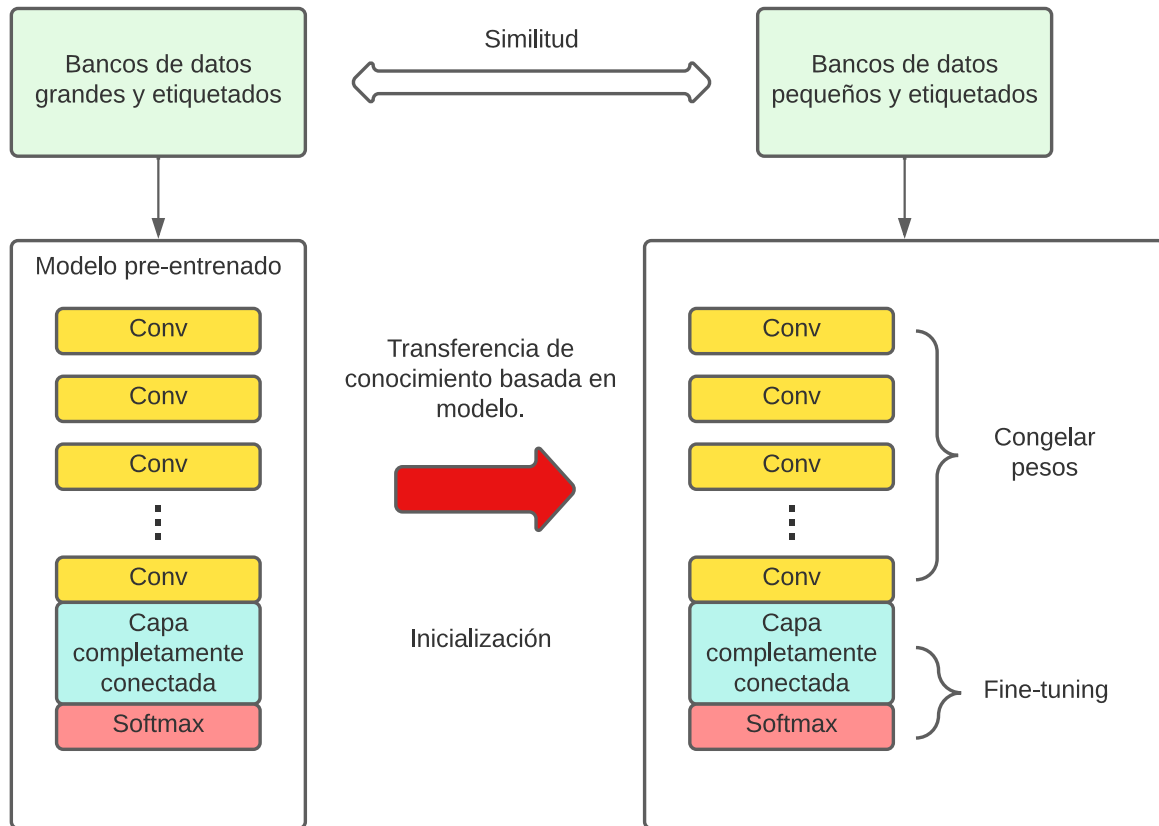


Figura 5.2: Diagrama explicativo del concepto de *fine-tuning*. (Fuente: Elaboración propia)

5.4 Resultados y discusión.

A continuación, se presentan los resultados obtenidos para distintas situaciones que consideran una distinta cantidad de datos disponibles, lo que se realiza para un entrenamiento aplicando técnicas TL y por otra parte un entrenamiento desde cero de la CNN.

5.4.1 Evaluación de la influencia en la cantidad de datos disponibles aplicando TL.

Los resultados obtenidos al aplicar TL para los bancos de 3 giros del eje y 10 giros del eje se presentan en las Figuras 5.4 y 5.5.

Gráficos de precisión y pérdida para distinta cantidad de datos disponibles.

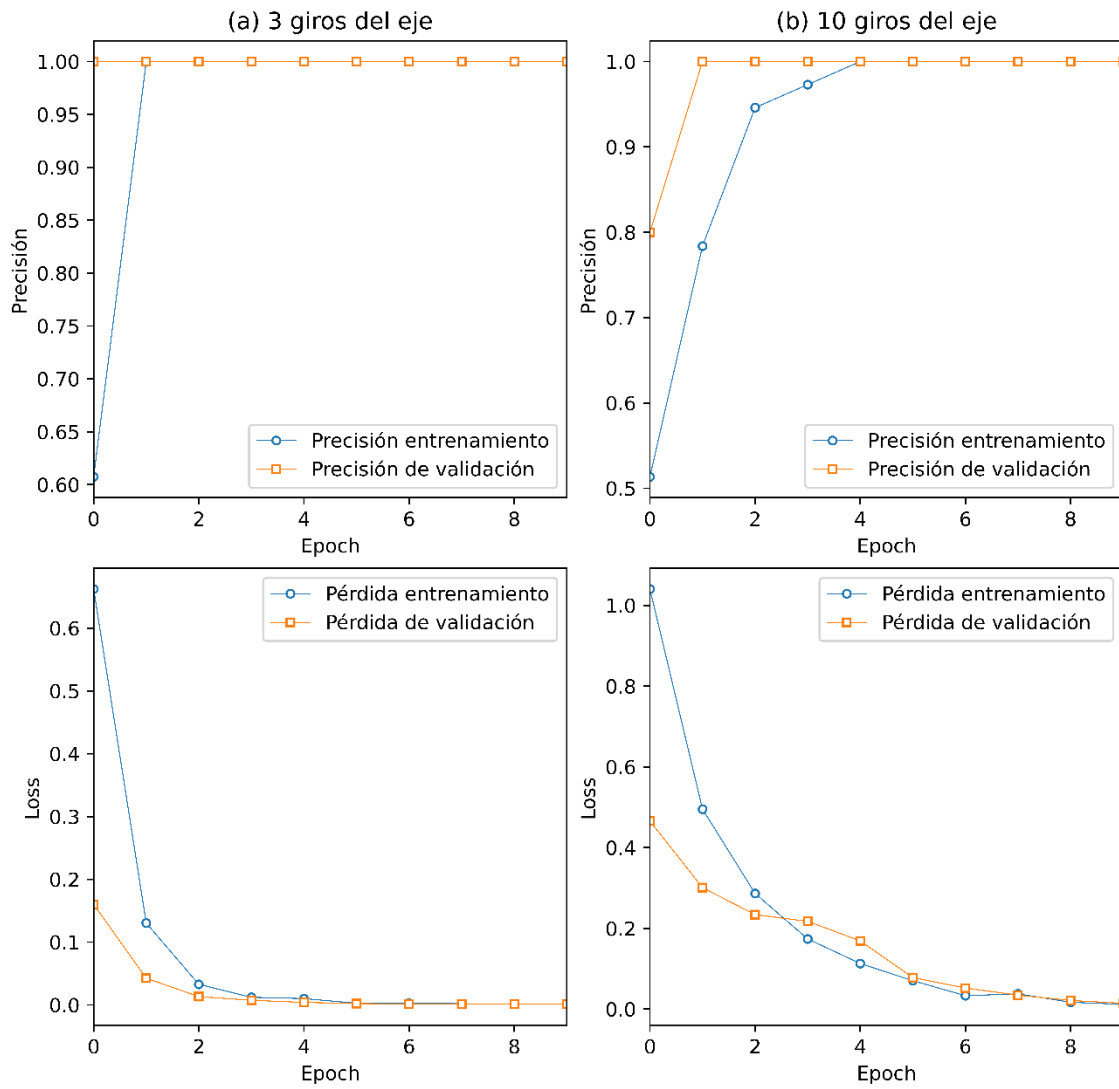


Figura 5.3: Gráficos de precisión y pérdida por épocas, (a) 3 giros del eje y (b) 10 giros del eje aplicando TL.
(Fuente: Elaboración propia)

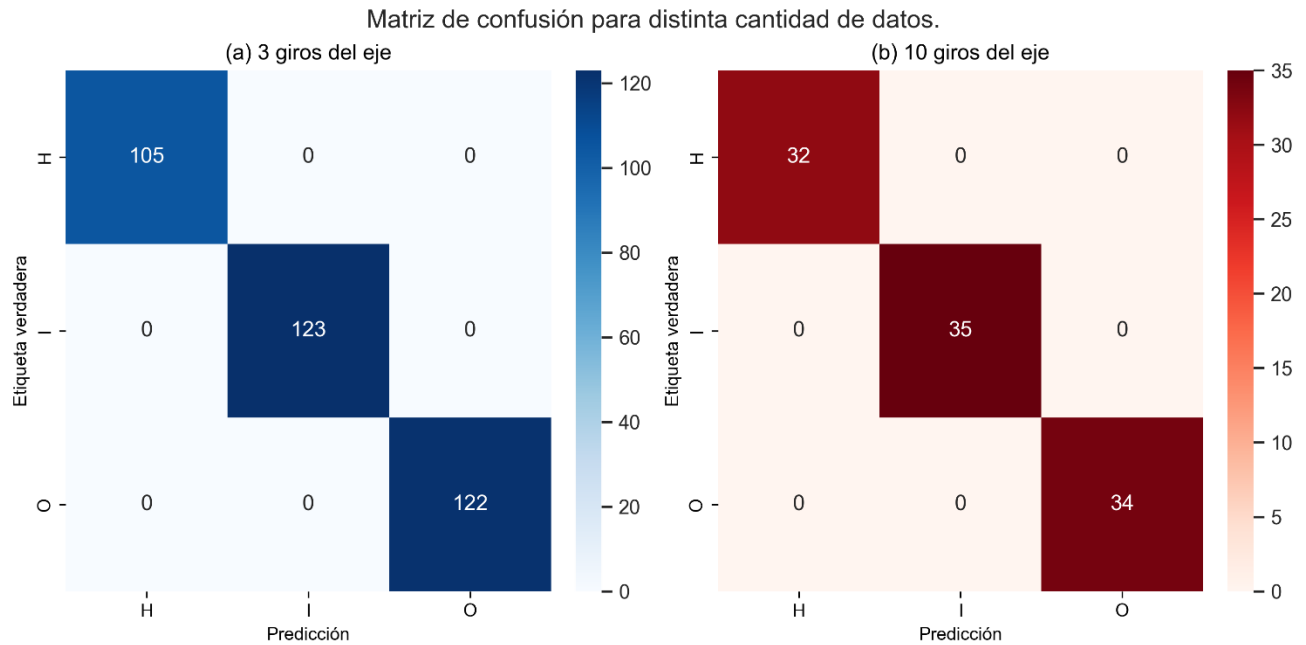


Figura 5.4: Matriz de confusión para datos de test, (a) 3 giros del eje y (b) 10 giros del eje aplicando TL. (Fuente: Elaboración propia)

De los gráficos de precisión y pérdida para el caso (a) 3 giros del eje, se observa una rápida convergencia a partir de la segunda época. Lo que implica que el uso de TL aumenta en gran medida la velocidad del entrenamiento, alcanzándose rápidamente precisiones del 100% para los datos de entrenamiento y validación. Situación similar ocurre para el caso (b) 10 giros del eje, el que alcanza precisiones del 100% en las épocas 4 y 5. Esta diferencia en la velocidad del entrenamiento se encuentra directamente relacionada con la cantidad de datos disponibles para el entrenamiento. Puesto que para el caso (a) se dispone de 135 datos para el entrenamiento y el caso (b) de tan solo 37 datos para su entrenamiento.

De la cantidad de datos disponibles, si bien existe a una diferencia significativa entre ellos. Se debe señalar que esta cantidad de datos por sí sola no bastarían para entrenar una arquitectura de DL desde cero, lo cual refleja la importancia y utilidad del uso de TL para aumentar la capacidad de generalización del modelo.

Por otra parte, los resultados que se obtienen en la matriz de confusión señalan una precisión del 100% para los datos de test de ambos casos (a) y (b). La cantidad de datos de test disponibles son de 350 para caso (a) 3 giros del eje y de 101 para el caso (b) 10 giros del eje. En donde para ambos casos, a pesar de la pequeña cantidad de datos de entrenamiento disponibles en comparación a los datos de test. Se tiene que el nuevo modelo entrenado aplicando *fine-tuning* y el conocimiento adquirido de un modelo pre-entrenado en otro dominio. Es capaz de reconocer las similitudes entre los modos de fallas de los distintos dominios y aumentar la robustez del nuevo modelo sin la necesidad de una gran cantidad de datos de entrenamiento.

5.4.2 Comparativa de transferencia de conocimiento y entrenamiento desde cero.

Para continuar el análisis de factibilidad del uso de TL, a continuación, se presentan los resultados que se obtienen al entrenar la arquitectura de CNN propuesta de cero.

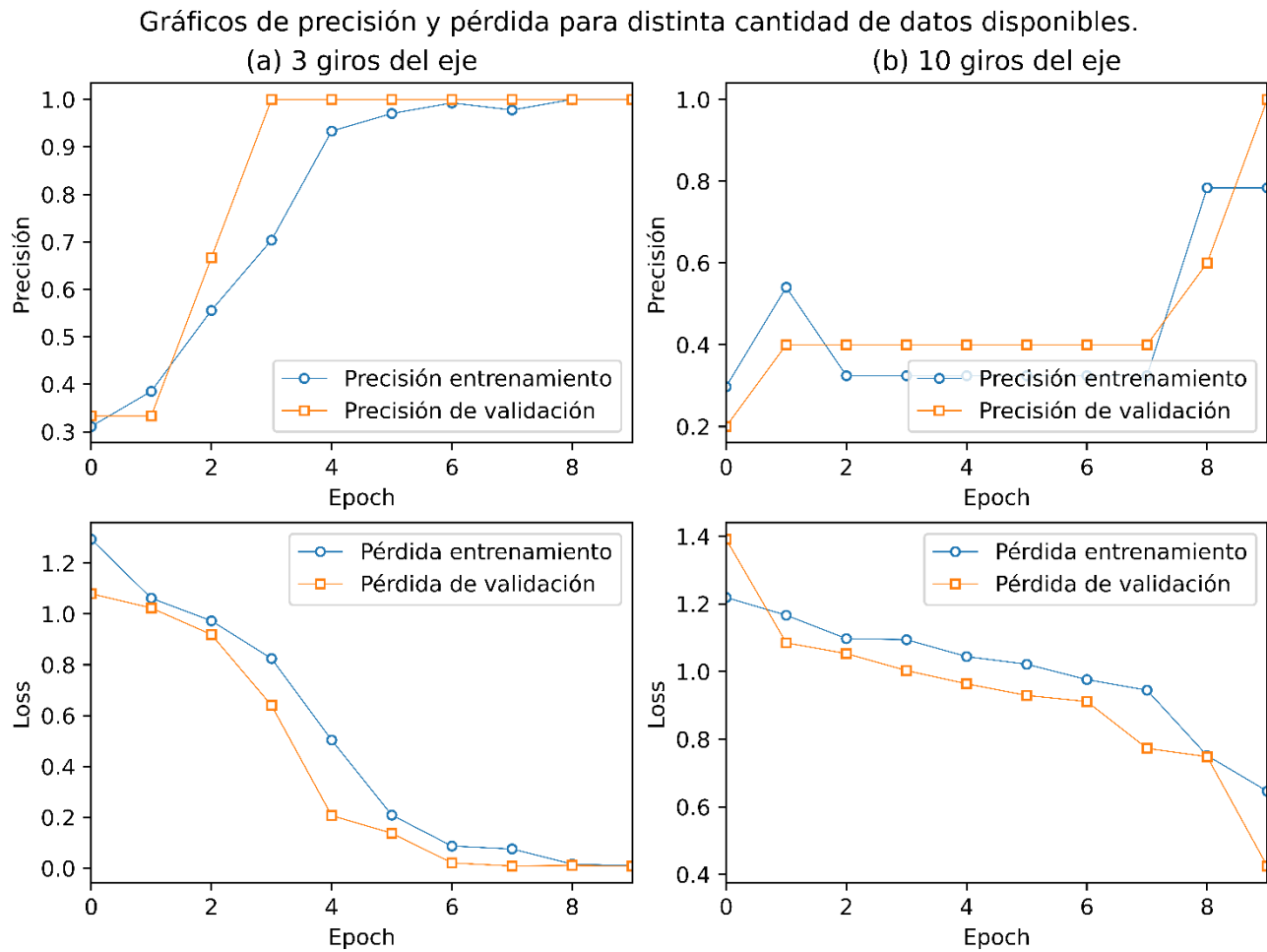


Figura 5.5: Gráficos de precisión y pérdida por épocas, (a) 3 giros del eje y (b) 10 giros del eje. (Fuente: Elaboración propia)

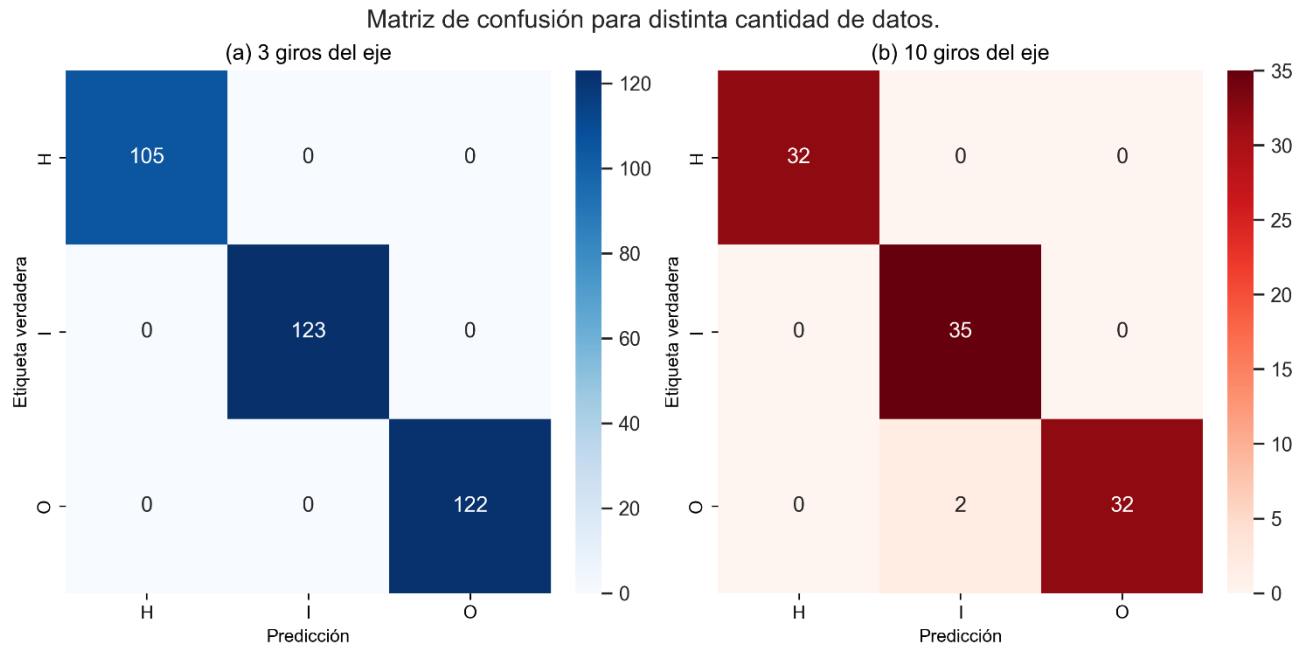


Figura 5.6: Matriz de confusión para datos de test, (a) 3 giros del eje y (b) 10 giros del eje. (Fuente: Elaboración propia)

De la Figura 5.6, se observa una clara diferencia entre los resultados aplicando TL y entrenar desde cero una CNN. Lo que se refleja en los valores de precisión y pérdida a través del paso de las épocas para el entrenamiento de la CNN desde cero. El que para el caso (a) 3 giros del eje alcanzan un 100% de precisión a partir de la época 9 y para el caso (b) 10 giros del eje, el modelo no alcanza a converger en las 10 épocas de entrenamiento. Para el caso (b) se obtiene una precisión del 78,38% de entrenamiento al final de la época 10.

En la matriz de confusión de cada caso presente en la Figura 5.6, se observa que el modelo es capaz de reconocer con 100% de precisión los modos de falla para el caso (a). Mientras que para el caso (b) se alcanza un 98,02% de precisión.

Los resultados obtenidos para el caso (b) demuestran las limitaciones del entrenamiento desde cero de la CNN. Requiriendo mayor número de épocas o cantidad de datos para obtener resultados parecidos a los obtenidos usando TL, con precisiones del 100% tanto para entrenamiento como test.

A continuación, en la Tabla 5.4 se presentan un resumen de los resultados obtenidos al iterar 10 veces los modelos para cada caso. Lo que se realiza con tal de evaluar la influencia en la inicialización de pesos en los resultados y obtener resultados más generales.

Tabla 5.4: Resultados después de 10 iteraciones para cada caso. (Fuente: Elaboración propia.).

Caso	Número de giros del eje	Máxima precisión	Mínima precisión	Precisión promedio	Desviación estándar
Entrenamiento desde cero.	3 giros	100 %	94 %	98,63 %	2,21
	10 giros	97,03 %	66,34 %	84,85 %	12,52
Transferencia de conocimiento	3 giros	100 %	100 %	100 %	0,00
	10 giros	100 %	95,05 %	97,43 %	1,99

De la Tabla 5.4, se obtiene que los métodos aplicando TL alcanzan una precisión del $100\% \pm 0,00$ para el banco de 3 giros del eje y un $97,42\% \pm 1,99$ para el banco de 10 giros del eje. Resultados superiores a los obtenidos entrenando la CNN desde cero, poniéndose en evidencia las ventajas de realizar TL. Alcanzándose mayores valores de precisión y una velocidad de entrenamiento más rápida, sin la necesidad de una gran cantidad de datos. Lo que supone una ventaja comparativa con respecto a su aplicación en situaciones reales de la industria.

5.4.3 Limitaciones actuales para la aplicación de transferencia de conocimiento.

Si bien las metodologías basadas en TL muestran resultados prometedores en resolver la falta de datos disponibles de falla para su entrenamiento. Estos métodos aún se encuentran limitados ante casos en donde no se tienen datos disponibles de falla, como es el caso de los motores de avión. Para lo anterior, aún se encuentran desarrollándose metodologías que contemplan la simulación y generación de datos artificiales de fallas que se asemejen a los reales, mediante por ejemplo modelos dinámicos de estos rodamientos en conjunto con el uso de una GAN para generar datos sintéticos [26].

Entre otras limitaciones asociadas a la aplicación de TL, se encuentran [23]:

- 1) La discrepancia en la distribución de los datos, la cual depende de factores como las condiciones de operación, causando que la distribución entre el dominio objetivo y fuente tienda a aumentar su discrepancia. Lo que tiene como consecuencia un peor desempeño del modelo objetivo.
- 2) Estos enfoques aún carecen de validación con datos reales aplicados a la industria, por lo que aún falta comprobar su capacidad de generalización para casos reales.
- 3) La generación de datos de falla artificiales, las cuales no se asemejan a las reales producidas comúnmente por degradación. Lo que significa contar con un monitoreo prolongado de esta condición para obtener datos para su entrenamiento y que el modelo sea capaz de diagnosticar fallas incipientes.

En resumen, al finalizar este capítulo se concluye que es factible el realizar TL respetándose las limitaciones aún presentes para el desarrollo de estos métodos. Las que para el caso de estudio utilizado no se presentaron dada la distribución similar entre los dominios seleccionados.

CAPÍTULO 6: Conclusiones y recomendaciones.

Entre los principales aportes de esta memoria se encuentra la evaluación de la influencia de distintas configuraciones como: la duración de señal, normalización y tamaño del ancho de ventana. Además, se realiza un análisis para evaluar la influencia de hiperparámetros de la CNN, como lo son las épocas y *learning rate*. De la evaluación de las configuraciones se obtiene que la cantidad de imágenes disponibles, intensidad de la escala de colores y tamaño de ancho de ventana, cumplen un rol fundamental para facilitar el entrenamiento de la CNN. Por otra parte, la correcta selección de hiperparámetros permite disminuir la variabilidad de resultados, en conjunto con mitigar la influencia asociada a la inicialización de pesos aleatoria de la CNN. Mediante el desarrollo de la metodología planteada se logra evaluar la influencia de estas configuraciones, y obtener como resultado un modelo de DL capaz de clasificar automáticamente imágenes tiempo-frecuencia asociadas a fallas en rodamientos. Este modelo es capaz de clasificar con una precisión de un 100% la falla presente en la imagen tiempo-frecuencia generada por la STFT. De manera que se cumple con lo propuesto inicialmente en la hipótesis de trabajo.

Por otra parte, también se investiga el principio de funcionamiento de la CNN mediante la visualización de sus características y su capacidad para extraer detalles asociados a cada tipo de falla. Lo que se realiza a través de la aplicación de los filtros aprendidos a las imágenes sobre las que el modelo es entrenado. De modo que la CNN es capaz de identificar características abstractas de los modos de fallas, previos a la entrada a la capa completamente conectada. En donde se refleja una mayor o menor activación de los filtros dependiendo del tipo de falla presente en la imagen y que corresponde a la manera en que la CNN reconoce estas.

En cuanto a la factibilidad del uso de TL, se obtienen resultados positivos en su aplicación a un nuevo caso de estudio. De donde se destaca la utilidad de esta herramienta, aumentando la capacidad de generalización, rapidez de convergencia y precisión del modelo. Además, de requerir de una menor cantidad de datos para el entrenamiento, lo que es favorable para su aplicación en un contexto industrial.

Entre las recomendaciones y trabajos futuros se encuentran la prueba de esta metodología para su aplicación en casos reales de la industria. También, se recomienda el uso de otras transformadas T-F, como la CWT y WVD, con tal de evaluar su influencia en el desempeño de la CNN. Lo que se puede complementar con distintas técnicas para aumentar la activación de los mapas de características. Entre estas técnicas se encuentran el cambio de la escala de normalización de colores a dB y uso de los tres canales RGB para el entrenamiento de la CNN. Por otra parte, también se propone el estudio de aplicación del modelo *Transformer* para la metodología presentada en reemplazo de la CNN, con tal de disminuir el costo computacional asociado a la implementación de estos modelos en la industria.

Finalmente, en el contexto del TL se recomienda explorar su aplicabilidad a distintos contextos industriales con tal de evaluar la influencia de la distribución de los datos disponibles. Lo que se complementarían con el desarrollo de una metodología para caso en los que no se tienen datos disponibles de falla.

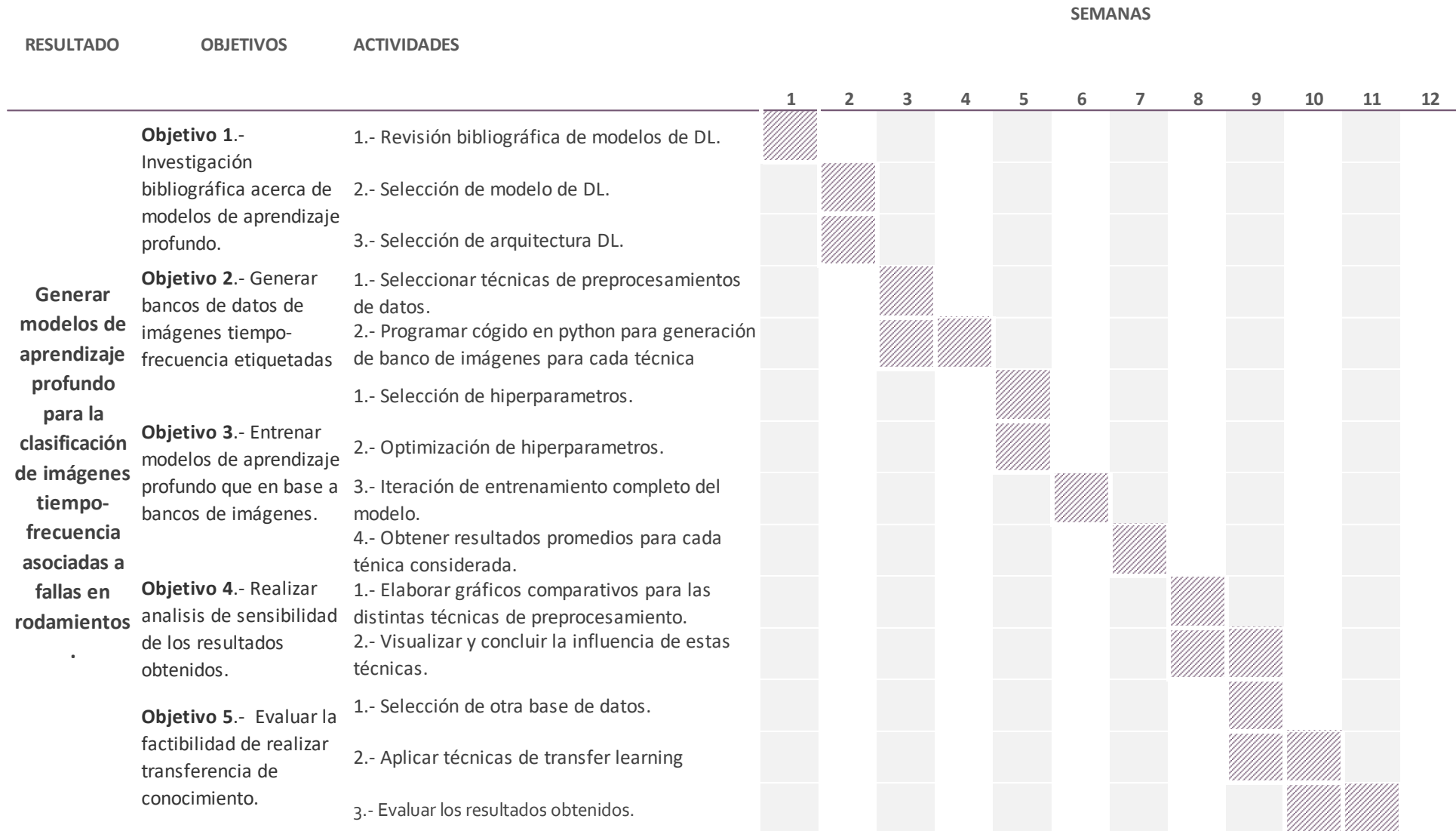
Referencias

- [1] J. Pacheco-Chérrez, J. A. Fortoul-Díaz, F. Cortés-Santacruz, L. María Alosó-Valerdi, and D. I. Ibarra-Zarate, “Bearing fault detection with vibration and acoustic signals: Comparison among different machine learning classification methods,” *Eng Fail Anal*, vol. 139, p. 106515, 2022, doi: <https://doi.org/10.1016/j.engfailanal.2022.106515>.
- [2] D. Neupane and J. Seok, “Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review,” *IEEE Access*, vol. 8, pp. 93155–93178, 2020, doi: [10.1109/ACCESS.2020.2990528](https://doi.org/10.1109/ACCESS.2020.2990528).
- [3] Z. Zhao, J. Wu, T. Li, C. Sun, R. Yan, and X. Chen, “Challenges and Opportunities of AI-Enabled Monitoring, Diagnosis & Prognosis: A Review,” *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, p. 56, 2021, doi: [10.1186/s10033-021-00570-7](https://doi.org/10.1186/s10033-021-00570-7).
- [4] R. Magar, L. Ghule, J. Li, Y. Zhao, and A. B. Farimani, “FaultNet: A Deep Convolutional Neural Network for Bearing Fault Classification,” *IEEE Access*, vol. 9, pp. 25189–25199, 2021, doi: [10.1109/ACCESS.2021.3056944](https://doi.org/10.1109/ACCESS.2021.3056944).
- [5] Z. Zhu *et al.*, “A review of the application of deep learning in intelligent fault diagnosis of rotating machinery,” *Measurement*, vol. 206, p. 112346, 2023, doi: <https://doi.org/10.1016/j.measurement.2022.112346>.
- [6] Y. Cheng, M. Lin, J. Wu, H. Zhu, and X. Shao, “Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network,” *Knowl Based Syst*, vol. 216, p. 106796, 2021, doi: <https://doi.org/10.1016/j.knosys.2021.106796>.
- [7] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, “Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016, doi: [10.1109/TIE.2016.2582729](https://doi.org/10.1109/TIE.2016.2582729).
- [8] G. Niu, X. Wang, M. Golda, S. Mastro, and B. Zhang, “An optimized adaptive PReLU-DBN for rolling element bearing fault diagnosis,” *Neurocomputing*, vol. 445, pp. 26–34, 2021, doi: <https://doi.org/10.1016/j.neucom.2021.02.078>.
- [9] K. Xu, X. Kong, Q. Wang, S. Yang, N. Huang, and J. Wang, “A bearing fault diagnosis method without fault data in new working condition combined dynamic model with deep learning,” *Advanced Engineering Informatics*, vol. 54, p. 101795, 2022, doi: <https://doi.org/10.1016/j.aei.2022.101795>.
- [10] Y. Ding, M. Jia, Q. Miao, and Y. Cao, “A novel time–frequency Transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings,” *Mech Syst Signal Process*, vol. 168, p. 108616, 2022, doi: <https://doi.org/10.1016/j.ymsp.2021.108616>.

- [11] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [12] M. S. Azari, F. Flammini, S. Santini, and M. Caporuscio, “A Systematic Literature Review on Transfer Learning for Predictive Maintenance in Industry 4.0,” *IEEE Access*, vol. 11, pp. 12887–12910, 2023, doi: 10.1109/ACCESS.2023.3239784.
- [13] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
- [14] Y. Zhang, K. Xing, R. Bai, D. Sun, and Z. Meng, “An enhanced convolutional neural network for bearing fault diagnosis based on time–frequency image,” *Measurement*, vol. 157, p. 107667, 2020, doi: <https://doi.org/10.1016/j.measurement.2020.107667>.
- [15] J. Wang, Z. Mo, H. Zhang, and Q. Miao, “A Deep Learning Method for Bearing Fault Diagnosis Based on Time-Frequency Image,” *IEEE Access*, vol. 7, pp. 42373–42383, 2019, doi: 10.1109/ACCESS.2019.2907131.
- [16] Q. Zhang and L. Deng, “An Intelligent Fault Diagnosis Method of Rolling Bearings Based on Short-Time Fourier Transform and Convolutional Neural Network,” *Journal of Failure Analysis and Prevention*, vol. 23, no. 2, pp. 795–811, 2023, doi: 10.1007/s11668-023-01616-9.
- [17] J. Wang, Z. Mo, H. Zhang, and Q. Miao, “A Deep Learning Method for Bearing Fault Diagnosis Based on Time-Frequency Image,” *IEEE Access*, vol. 7, pp. 42373–42383, 2019, doi: 10.1109/ACCESS.2019.2907131.
- [18] C. T. Alexakos, Y. L. Karnavas, M. Drakaki, and I. A. Tziafettas, “A Combined Short Time Fourier Transform and Image Classification Transformer Model for Rolling Element Bearings Fault Diagnosis in Electric Motors,” *Mach Learn Knowl Extr*, vol. 3, no. 1, pp. 228–242, 2021, doi: 10.3390/make3010011.
- [19] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, “Deep Learning Enabled Fault Diagnosis Using Time-Frequency Image Analysis of Rolling Element Bearings,” *Shock and Vibration*, vol. 2017, p. 5067651, 2017, doi: 10.1155/2017/5067651.
- [20] H. Liu, L. Li, and J. Ma, “Rolling Bearing Fault Diagnosis Based on STFT-Deep Learning and Sound Signals,” *Shock and Vibration*, vol. 2016, p. 6127479, 2016, doi: 10.1155/2016/6127479.

- [21] H. Huang and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions," *Data Brief*, vol. 21, pp. 1745–1749, 2018, doi: <https://doi.org/10.1016/j.dib.2018.11.019>.
- [22] M. Kim, J. H. Jung, J. U. Ko, H. B. Kong, J. Lee, and B. D. Youn, "Direct Connection-Based Convolutional Neural Network (DC-CNN) for Fault Diagnosis of Rotor Systems," *IEEE Access*, vol. 8, pp. 172043–172056, 2020, doi: [10.1109/ACCESS.2020.3024544](https://doi.org/10.1109/ACCESS.2020.3024544).
- [23] M. Hakim, A. A. B. Omran, A. N. Ahmed, M. Al-Waily, and A. Abdellatif, "A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations," *Ain Shams Engineering Journal*, vol. 14, no. 4, p. 101945, 2023, doi: <https://doi.org/10.1016/j.asej.2022.101945>.
- [24] Z. Chen, J. Cen, and J. Xiong, "Rolling Bearing Fault Diagnosis Using Time-Frequency Analysis and Deep Transfer Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 150248–150261, 2020, doi: [10.1109/ACCESS.2020.3016888](https://doi.org/10.1109/ACCESS.2020.3016888).
- [25] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: [10.1109/JPROC.2020.3004555](https://doi.org/10.1109/JPROC.2020.3004555).
- [26] W. Hou, C. Zhang, Y. Jiang, K. Cai, Y. Wang, and N. Li, "A new bearing fault diagnosis method via simulation data driving transfer learning without target fault data," *Measurement*, vol. 215, p. 112879, 2023, doi: <https://doi.org/10.1016/j.measurement.2023.112879>.
- [27] "Fault Data Sets - Society For Machinery Failure Prevention Technology." <https://www.mfpt.org/fault-data-sets/> (accessed Jun. 07, 2023).

Anexo A: Planificación Carta Gantt.



Anexo A2: Github.

Las actualizaciones y desarrollo de esta memoria se publicarán en:
<https://github.com/crisbrz/Memoria-de-Titulo>

Anexo A3: Métricas de precisión, Recall, F1-Score.

Términos y definiciones

Estas métricas dentro del campo de ML se utilizan para obtener una idea detallada de como de desempeña el clasificador. Lo que se realiza en lugar de observar solamente la precisión general del modelo.

A continuación, se presentan algunos términos necesarios para comprender el significado de estas métricas:

- 1) **Positivo:** Este término se utiliza cuando la instancia predicha es clasificada como un miembro de lo que está tratando de identificar con el clasificador. Por ejemplo, un clasificador de imágenes de gatos clasificara positivamente las imágenes de gatos cuando esta sea correcta.
- 2) **Negativo:** Se define cuando la instancia predicha es clasificada como un miembro que no pertenece a la clase que está intentando clasificar. Por ejemplo, un clasificador de gatos debería clasificar fotos que tengan perros como una instancia negativa.
- 3) **Verdadero positivo o True positive (TP):** La predicción del modelo es positiva y el valor verdadero o real de la instancia también es positiva.
- 4) **Verdadero negativo o True negative (NP):** La predicción del modelo es negativa y el valor verdadero o real de la instancia también es negativa.
- 5) **Falso positivos o False positive (FP):** La predicción del modelo es positiva, pero el valor verdadero o real de la instancia es negativa.
- 6) **Falso negativo o False negative (FN):** La predicción del modelo es negativa, pero el valor verdadero o real de la instancia es positiva.

Definiciones de Accuracy, Precision, Recall y F1-Score.

En base a los términos anteriores se definen las siguientes métricas:

- 1) **Precisión o Accuracy:** Mide el porcentaje de casos que el modelo acierta. El Accuracy se calcula como:

$$Accuracy = \frac{Predicciones\ correctas}{Total\ de\ predicciones} \quad (12)$$

- 2) **Precision:** Determina que fracción de las predicciones indicadas como positivas son realmente positivas. La precisión se calcula como:

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

3) *Recall*: Mide la capacidad del modelo para predecir positivos. El recall se calcula como:

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

4) *F1-Score*: El F1-Score considera los resultados de las métricas de *precision* y *recall*. El F1-Score se calcula como:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (15)$$

Ejemplo de cálculo de métricas para multiclases.

Para ejemplificar el proceso de cálculo de las métricas mencionadas se considera la siguiente matriz de confusión recuperada de la Sección 4.1.3.

Matriz de confusión para datos de test - 10 giros por eje.

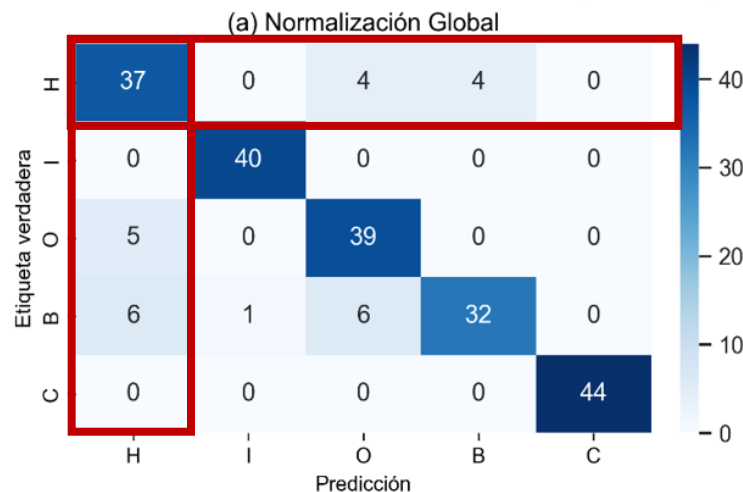


Figura A3.1: Matriz de confusión para datos de test 10 giros del eje y normalización global. (Fuente: Elaboración propia)

Para la clase con condición normal (H), las métricas se calculan mediante el siguiente proceso:

- 1) *Precision*: Las predicciones positivas y que del conjunto de datos corresponden realmente a positivos son de 37 instancias, lo que se considera como el valor para el TP. Por otra parte, las instancias que fueron predichas como FN son de un total 11. Puesto que están fueron predichas como otras clases siendo que correspondían a la clase H. De modo que se tiene:

$$Precision = \frac{TP}{TP + FP} = \frac{37}{37 + 11} = 0,770 \quad (15)$$

- 2) *Recall*: Los falsos negativos según la definición corresponden a los indicados en el rectángulo horizontal, siendo de un total de 8 FN. De modo que se tiene:

$$Recall = \frac{TP}{TP + FN} = \frac{37}{37 + 8} = 0,822 \quad (16)$$

- 3) *F1-Score*: Una vez ya calculados los valores para *Precision* y *Recall* se tiene que el valor para esta métrica es de:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0,770 \cdot 0,822}{0,770 + 0,822} = 0,795 \quad (17)$$

Es proceso de cálculo se realiza para cada clase, obteniendo así los resultados que se presentaron en la Tabla 4.2 y Tabla 4.4.