

**PONTIFICIA UNIVERSIDAD
CATÓLICA DEL PERÚ**

Escuela de Posgrado



Aplicación de un modelo de Deep Learning con redes neuronales convolucionales para evaluar la integridad estructural y predecir fallas en puentes de concreto mediante exploración visual de imágenes.

Tesis para obtener el grado académico de Maestro en Ingeniería Civil con mención en Diagnóstico y Reparación de Construcciones Patrimoniales y Existentes que presenta:

Richard Frank Rojas Atachao

Asesor:

José Alberto Acero Martínez

Lima, 2025

Informe de Similitud

Yo, José Alberto Acero Martínez, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado “Aplicación de un modelo de Deep Learning con redes neuronales convolucionales para evaluar la integridad estructural y predecir fallas en puentes de concreto mediante exploración visual de imágenes”, del/de la autor(a) / de los(as) autores(as):


Richard Frank Rojas Atachao

Dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 16%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 21/02/2025.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

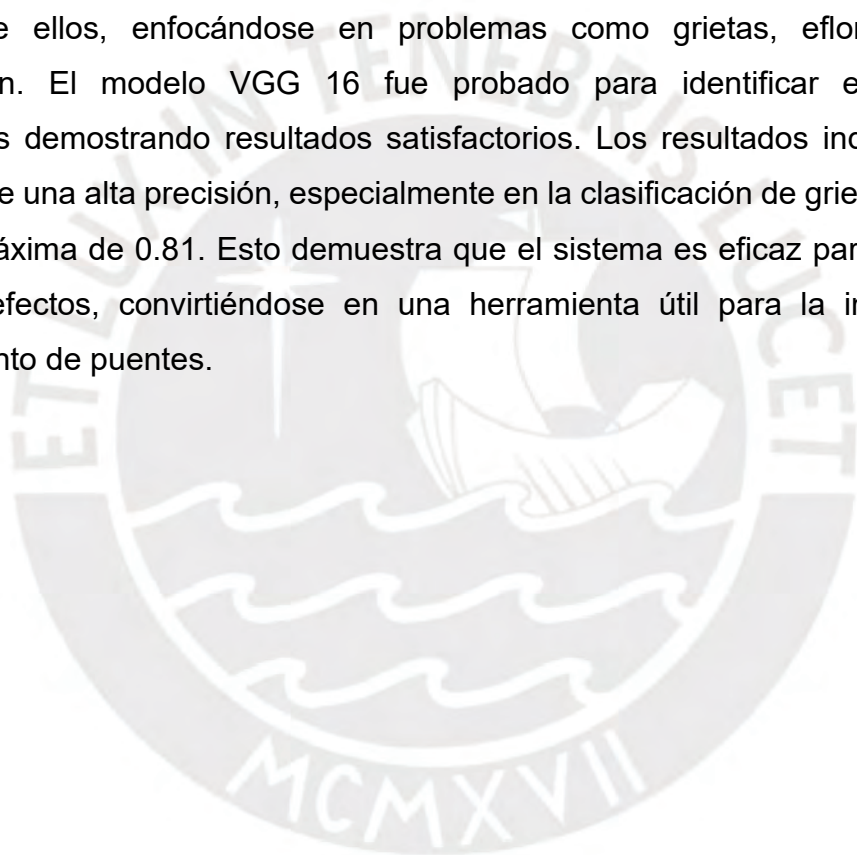
Lugar y fecha:

Lima 25 de febrero del 2025.

JOSE ALBERO ACERO MARTINEZ	
DNI: 29662895	 FIRMA:
ORCID: 0000-0003-4154-9510	

RESUMEN

En el área de la ingeniería civil, es muy importante revisar la salud estructural de los puentes de concreto. Esta revisión tiene el objetivo de mantener la seguridad y mejorar el mantenimiento de estas estructuras. Por otro lado, en el área de la Inteligencia Artificial, las Redes Neuronales Convolucionales se han usado con éxito para analizar y clasificar imágenes de distintos tipos y orígenes. Esta tesis tiene como objetivo crear un sistema que use un modelo de Aprendizaje Profundo usando Redes Neuronales Convolucionales con la arquitectura VGG 16. Su función principal es evaluar la estructura de los puentes y clasificar los daños a través de imágenes tomadas de ellos, enfocándose en problemas como grietas, eflorescencia y descamación. El modelo VGG 16 fue probado para identificar estos daños estructurales demostrando resultados satisfactorios. Los resultados indican que el sistema tiene una alta precisión, especialmente en la clasificación de grietas, con una precisión máxima de 0.81. Esto demuestra que el sistema es eficaz para detectar y clasificar defectos, convirtiéndose en una herramienta útil para la inspección y mantenimiento de puentes.



ABSTRACT

In the area of civil engineering, it is very important to review the structural health of concrete bridges. This review aims to maintain the safety and improve the maintenance of these structures. On the other hand, in the area of Artificial Intelligence, Convolutional Neural Networks have been successfully used to analyze and classify images of different types and origins. This thesis aims to create a system that uses a Deep Learning model using Convolutional Neural Networks with the VGG 16 architecture. Its main function is to evaluate the structure of bridges and classify damages through images taken from them, focusing on problems such as cracks, efflorescence and peeling. The VGG 16 model was tested to identify these structural damages demonstrating satisfactory results. The results indicate that the system has a high accuracy, especially in the classification of cracks, with a maximum accuracy of 0.81. This shows that the system is effective in detecting and classifying defects, becoming a useful tool for the inspection and maintenance of bridges.

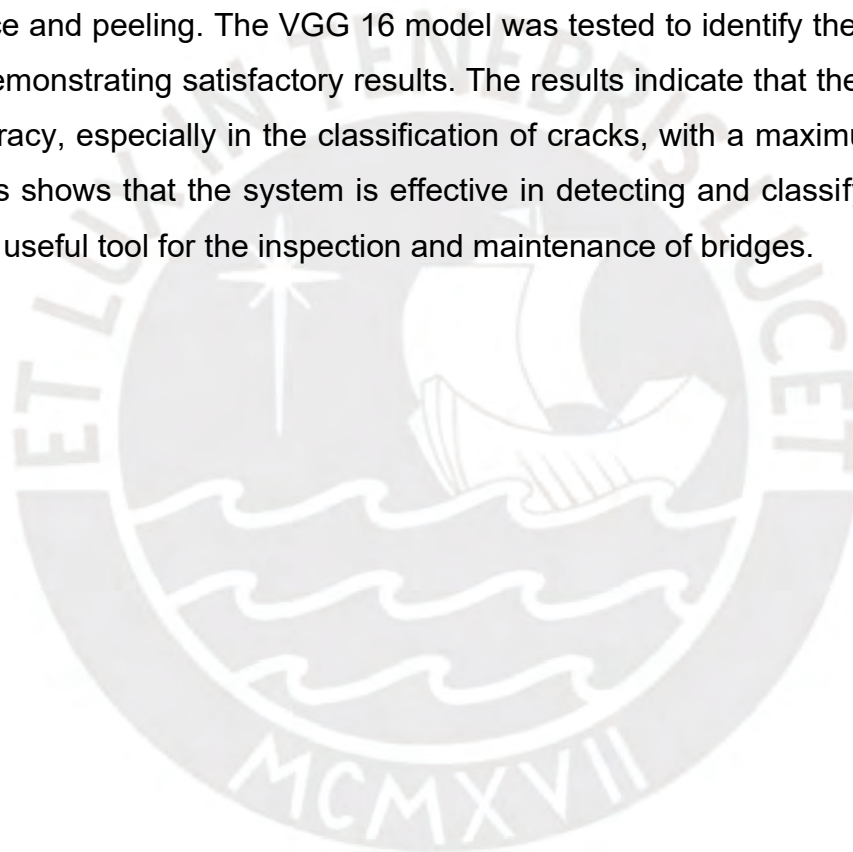
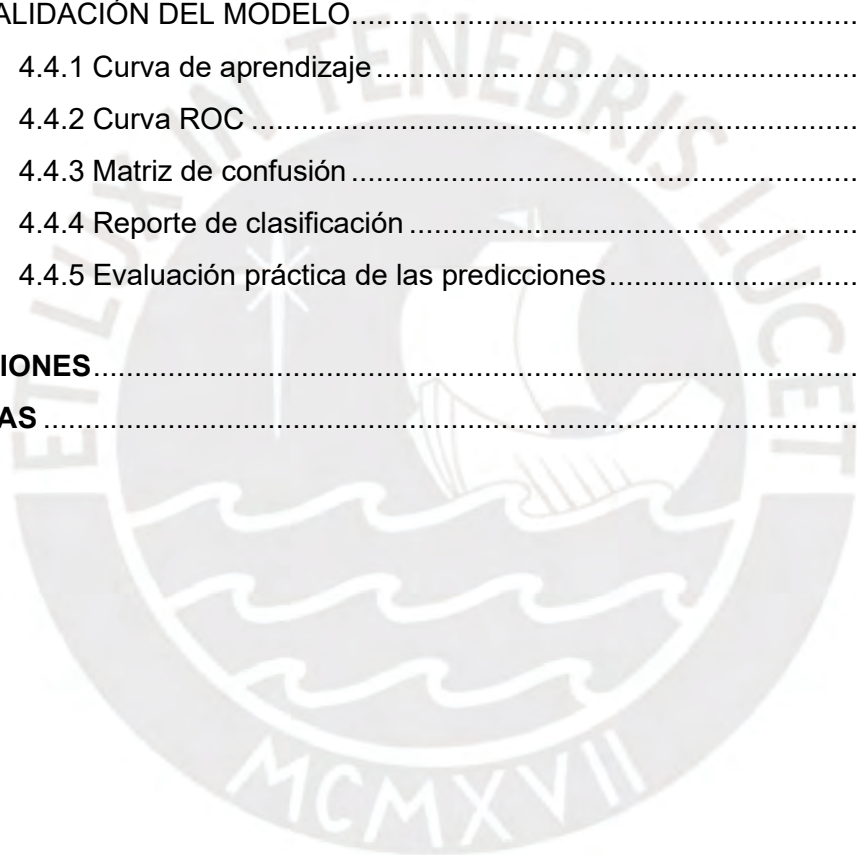


TABLA DE CONTENIDO

RESUMEN	3
ABSTRACT	3
I.GENERALIDADES	7
1.1 INTRODUCCION.....	7
1.2 ANTECEDENTES	9
1.3 JUSTIFICACION	14
1.4 OBJETIVOS	15
1.4.1 Objetivo General	15
1.4.2 Objetivos Específicos.....	15
1.5 ALCANCES Y LIMITACIONES.....	16
1.5.1 Alcances	16
1.5.2 Limitaciones.....	16
1.6 HIPÓTESIS	17
1.6.1 Hipótesis General.....	17
1.6.2 Hipótesis Específicas	17
II.MARCO TEÓRICO	18
2.1 INTRODUCCION TEÓRICA.....	18
2.2 INGENIERÍA ESTRUCTURAL.....	18
2.2.1 Puentes.....	19
2.2.2 Patologías en puentes	22
2.2.3 Origen de las patologías en puentes.....	26
2.3 APRENDIZAJE DE MÁQUINA	28
2.3.1 Aprendizaje supervisado	29
2.3.2 Aprendizaje no supervisado	30
2.3.3 Aprendizaje por refuerzo.....	31
2.3.4 Aprendizaje profundo	32
2.4 PROCESAMIENTO DIGITAL DE IMÁGENES.....	34
2.5 SISTEMAS DE MONITOREO DE SALUD ESTRUCTURAL EN PUENTES	36
III.METODOLOGÍA	38
3.1 RECOLECCIÓN DE DATOS	38
3.1.1 Recopilación de datos.....	38
3.2 PREPARACIÓN Y LIMPIEZA DE DATOS	39

3.3 ENTRENAMIENTO Y EVALUACIÓN DEL MODELO	40
3.3.1 Entrenamiento de la CNN	40
3.3.2 Descripción de la CNN	41
3.3.3 Evaluación de la CNN	44
3.4 USO POTENCIAL DE UN BHMS	47
IV.RESULTADOS	49
4.1 RECOLECCIÓN DE DATOS	49
4.1.1 Recopilación de datos	49
4.2 SELECCIÓN Y PREPARACIÓN DE LAS IMÁGENES.....	49
4.3 ENTRENAMIENTO DEL MODELO	51
4.4 VALIDACIÓN DEL MODELO.....	53
4.4.1 Curva de aprendizaje	54
4.4.2 Curva ROC	56
4.4.3 Matriz de confusión	59
4.4.4 Reporte de clasificación	61
4.4.5 Evaluación práctica de las predicciones.....	63
V.CONCLUSIONES	66
REFERENCIAS	68



I. GENERALIDADES

1.1 INTRODUCCIÓN

En ingeniería civil, especialmente en la gestión de infraestructuras, es crucial evaluar la integridad estructural de los puentes de concreto. También es importante clasificar las fallas en estos puentes. Estas tareas son esenciales para garantizar la seguridad pública y mejorar el mantenimiento de estos elementos clave (He et al., 2022). Tradicionalmente, las evaluaciones se hacen con inspecciones visuales manuales por expertos. Este proceso puede ser caro, subjetivo y con riesgo de errores humanos (Sun et al., 2020). Sin embargo, los avances en tecnología, especialmente en Inteligencia Artificial (IA) y herramientas como el Aprendizaje Profundo (DL, por sus siglas en inglés), ofrecen nuevas formas de hacer estas inspecciones más precisas y eficientes (Mishra et al., 2022).

Las herramientas del DL como las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés), han demostrado un notable éxito en el análisis y clasificación de imágenes en diversos campos (Flotzinger et al., 2023). Las CNNs tienen la capacidad de aprender características complejas haciendo uso de grandes volúmenes de datos visuales, i.e. imágenes. Esto último convierte a las CNNs en instrumentos o herramientas idóneas para la evaluación automática de imágenes de infraestructuras de puentes. Aplicar CNNs en la evaluación de puentes no sólo puede mejorar la detección de daños estructurales en puentes de concreto, sino también facilitar la detección temprana de fallas; permitiendo así intervenciones proactivas y reduciendo el riesgo de incidentes catastróficos.

Esta tesis se enfoca en el proyecto de desarrollar un sistema utilizando un modelo de DL con una CNN basada en VGG 16 e implementada en Python. La Figura 1 muestra un esquema del sistema propuesto. El objetivo final es evaluar la salud o integridad estructural y clasificar patologías o fallas en puentes mediante la exploración visual de imágenes capturadas de estas estructuras. El enfoque basado en CNNs permite automatizar el proceso de inspección, proporcionando una herramienta potencialmente útil y eficiente para ingenieros y gestores de infraestructuras.

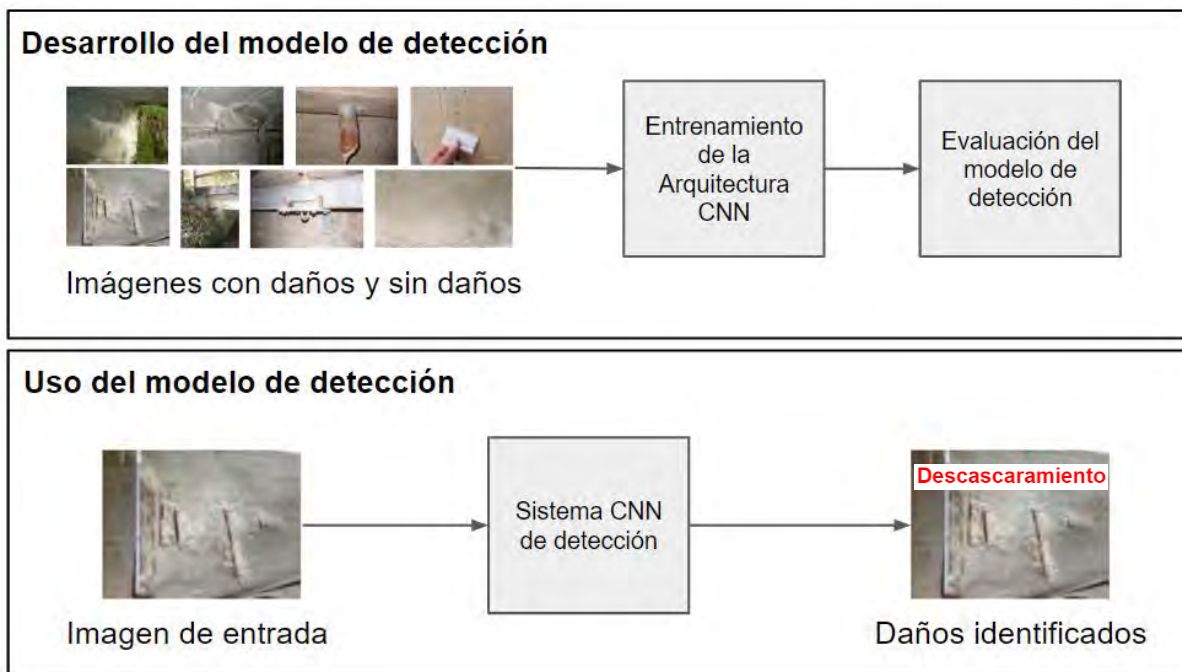


Figura 1. Propuesta de sistema para la evaluación de la salud estructural de puentes de concreto.

El lenguaje de programación usado en este proyecto es Python. Su uso se justifica por su versatilidad y la amplia disponibilidad de bibliotecas especializadas en aprendizaje profundo, como TensorFlow y Keras. Estas herramientas permiten la implementación de modelos complejos de manera relativamente sencilla, facilitando la experimentación y optimización de las redes neuronales. Además, Python ofrece robustas capacidades relacionadas con el procesamiento de imágenes, esenciales tanto para el preprocesamiento como para el procesamiento de datos visuales (imágenes) y la mejora de la precisión del modelo de clasificación.

Este proyecto busca aportar una solución a la ingeniería civil y mejorar el uso de tecnologías de IA en la gestión de infraestructuras críticas, como los puentes de concreto. El sistema desarrollado en esta tesis ofrece una base para futuras investigaciones y desarrollos en este campo, lo que tiene el potencial de mejorar la capacidad de monitoreo y mantenimiento preventivo de puentes y estructuras similares.

El resto del documento sigue el siguiente orden, el Capítulo II, Marco Teórico, presenta los elementos teóricos necesarios para comprender tanto la propuesta del proyecto de tesis como los resultados. El Capítulo III, Metodología, describe el proceso seguido para la realización del presente proyecto de tesis. El Capítulo IV, Resultados, describe con detalle el rendimiento del sistema propuesto. Por último, el Capítulo V, presenta las conclusiones de la presente tesis.

1.2 ANTECEDENTES

En este apartado, se describe el trabajo relacionado con el proyecto de tesis con el fin de contextualizar el problema. Se efectúa una descripción de diferentes trabajos, y se describe cómo éstos se relacionan con la presente tesis.

El trabajo de Flah et al. (2021) habla sobre el uso de algoritmos de Machine Learning (ML) en el Monitoreo de la Salud Estructural (SHM). Destaca cómo estos algoritmos ayudan a detectar daños en estructuras civiles. El artículo menciona que, con el avance de tecnologías como el Internet de las Cosas y el big data, el uso de estas herramientas crecerá rápidamente. El ML puede hacer análisis eficientes de daños en diferentes estructuras, como puentes, edificios, presas y turbinas eólicas. El estudio clasifica los algoritmos de ML en dos áreas principales: SHM basado en vibraciones y SHM basado en imágenes. Se analiza cómo estos algoritmos funcionan y se dan recomendaciones para investigaciones futuras. Este artículo es útil para este proyecto, ya que ofrece una base sobre el uso de Machine Learning en el monitoreo estructural, especialmente en la inspección visual de imágenes.

El trabajo de Svendsen et al. (2022) habla sobre la necesidad de sistemas confiables para monitorear la salud estructural de puentes. En su estudio, se obtienen datos de un puente real en diferentes condiciones, simulando daños comunes en puentes de acero. Se usan dos grupos de sensores para capturar las respuestas locales y globales del puente. Luego, se aplica un algoritmo de ML no supervisado y se compara con cuatro algoritmos de ML supervisados. Se evalúa qué tipo de daños se pueden detectar mejor con estos algoritmos. Los resultados muestran que el aprendizaje automático no supervisado puede detectar daños casi tan bien como el

supervisado. Este estudio es importante porque ayuda a establecer una metodología para detectar daños en puentes de acero. También es relevante para este proyecto, ya que busca detectar daños en puentes de acero utilizando aprendizaje automático.

De forma similar, Bao y Li (2021) destacan que se ha investigado mucho sobre la metodología de monitoreo de la salud estructural basada en vibraciones. Sin embargo, los métodos convencionales tienen dificultades para detectar daños en estructuras reales. Esto se debe a la falta de información completa en el monitoreo. También a las intensas incertidumbres en las condiciones estructurales y los sistemas de monitoreo. Además, los efectos combinados del daño y las condiciones ambientales afectan los parámetros modales. Este trabajo presenta un marco conceptual importante sobre cómo el aprendizaje automático puede ser aplicado para evaluar y monitorear la salud estructural. Además, el artículo brinda una visión general de los principios y desafíos en el desarrollo de metodologías de diagnóstico y pronóstico de la salud estructural basadas en el aprendizaje automático.

Otro estudio que aborda el uso de técnicas de supervisión de la condición estructural en puentes es el publicado por Sun et al., (2020). En el trabajo, los autores mencionan que debido a las limitaciones en capacidad computacional y análisis de datos, no se aprovechan bien los datos masivos de SHM. También señalan que el uso de Big Data (BD) e IA es una forma prometedora de resolver este problema. El artículo explica cómo estas técnicas pueden aplicarse al SHM de puentes, resumiendo las técnicas existentes y los requisitos para nuevas aplicaciones. Este estudio es relevante para este proyecto de tesis, ya que proporciona una revisión detallada de cómo se están utilizando las técnicas de IA para supervisar la condición estructural de puentes, y ofrece una visión clara de las posibilidades y retos en el monitoreo de puentes grandes.

Otro trabajo de interés para el presente proyecto de tesis es el publicado por Malekloo et al., (2022). El artículo revisa cómo las técnicas tradicionales de detección de daños están siendo reemplazadas por soluciones inteligentes de monitoreo en tiempo real en los sistemas de salud estructural, destacando el papel clave del ML en este cambio. El artículo se enfoca en la integración del Internet de las Cosas (IoT) y el análisis de grandes volúmenes de datos (BD) para mejorar la capacidad de los

sistemas de monitoreo. Además, el trabajo proporciona una visión detallada de los avances recientes en la combinación de SHM y ML. También menciona el potencial de tecnologías emergentes como dispositivos móviles, drones, realidad virtual/aumentada y gemelos digitales para mejorar la eficiencia y precisión en el monitoreo de infraestructuras. Este trabajo identifica desafíos actuales y futuros en la unión de SHM y ML, y ofrece una perspectiva sobre el futuro de estos sistemas en la evaluación de la infraestructura civil.

Yuan et al. (2020), por su parte, publicaron un artículo que habla sobre las limitaciones del enfoque físico para el monitoreo de salud estructural, especialmente en estructuras complejas en lugares sin control. Explican cómo las nuevas tecnologías y sensores permiten medir muchos parámetros en estructuras reales usando redes de sensores. También mencionan que, en lugar de usar un enfoque físico, ahora se está utilizando más el enfoque basado en datos para SHM, pero aún falta información sobre daños en diferentes escenarios. Proponen el aprendizaje informado por la física como una posible solución, combinando el conocimiento técnico con el aprendizaje automático. Este enfoque está relacionado con el proyecto de tesis de este documento, que explora cómo integrar el conocimiento del área en la inteligencia artificial, para mejorar la precisión y el rendimiento del modelo de aprendizaje profundo (DL), que se usa para evaluar la salud de puentes de concreto y predecir fallos a través de imágenes visuales.

Sarmadi y Yuen, (2022). En su artículo, examinan las limitaciones del enfoque basado en física para el monitoreo de salud estructural, especialmente en estructuras complejas en entornos no controlados. Destacan cómo los avances en tecnología y sensores permiten monitorear muchos parámetros en estructuras reales mediante redes de sensores in situ. Proponen el aprendizaje informado por la física como una posible solución, integrando conocimientos técnicos en el proceso de aprendizaje. Este enfoque está relacionado con el proyecto de tesis, ya que explora cómo incluir este conocimiento en el aprendizaje automático.

Por su parte, He et al., (2022) investigaron el monitoreo estructural integrado. Este enfoque combina análisis de mecanismos, tecnología de monitoreo y análisis de datos. Su objetivo es detectar y clasificar daños en las estructuras. Esto incluye daños

repentinos o acumulativos. Todo esto busca garantizar el buen funcionamiento de los puentes. Los sistemas de monitoreo integrado han mejorado el mantenimiento y gestión de puentes. Esto se logra mediante monitoreo continuo y evaluaciones frecuentes de las condiciones de trabajo. El artículo analiza las tendencias actuales y futuras en el monitoreo de puentes. Se enfoca en tecnologías avanzadas, transmisión y análisis de datos de sensores, y modelos de predicción y alerta temprana. Se revisan y comparan cuatro tecnologías comunes de sensores: fibra óptica, piezoeléctrico, GNSS y magnetostrictivos. También se analizan métodos de transmisión de datos inalámbrica como ZigBee, Bluetooth, NB-IoT, Wi-Fi y LoRa. Se presentan métodos de procesamiento de datos con inteligencia artificial y sistemas de predicción de rendimiento y alerta temprana. Este artículo se relaciona con el proyecto de tesis, ya que también se investiga el uso de una CNN para evaluar la integridad estructural y predecir fallas en puentes usando imágenes visuales. Ofrece una visión detallada de las tecnologías y metodologías del monitoreo estructural en puentes.

Es de interés también el trabajo presentado por Bao et al., (2020). En su artículo, los autores proponen un enfoque de aprendizaje automático para la reconstrucción de datos. Este tema se usa en el monitoreo estructural para obtener y reconstruir datos, transmitirlos de forma inalámbrica e identificar daños en estructuras. La reconstrucción se plantea como una tarea de aprendizaje supervisado, utilizando el conocimiento previo como entrada y objetivo de la red. Los autores muestran la efectividad de este enfoque con ejemplos numéricos y pruebas de campo en un puente colgante. Este artículo se relaciona con la tesis, ya que trata sobre técnicas de reconstrucción de datos. Esto es importante para evaluar la integridad estructural y predecir fallas en puentes usando imágenes visuales.

En su artículo, Gardner et al. (2020) comparan la evaluación no destructiva (NDE) y el monitoreo de la salud estructural. Se enfocan en la inspección ultrasónica por ondas guiadas como tecnología común. En el artículo, los autores hablan de cómo el análisis de datos y el aprendizaje automático mejoran estas técnicas. Explican los algoritmos disponibles y cómo manejar grandes cantidades de datos. También presentan estudios de caso sobre monitoreo de estructuras. Mencionan los retos de los datos complejos, como en la inspección robótica. Este artículo es relevante para el proyecto

de tesis, ya que muestra cómo aplicar aprendizaje automático a la inspección ultrasónica y al monitoreo estructural. Ayuda a desarrollar modelos de aprendizaje profundo para evaluar puentes y predecir fallas usando imágenes. También explica cómo manejar los grandes datos de monitoreo.

El artículo de Ortigoza Quiroga, (s.f) estudia la inspección visual con drones para detectar patologías en construcciones de difícil acceso. Analiza los beneficios, limitaciones y desafíos de esta tecnología. Revisa la literatura, casos prácticos y compara metodologías en términos de precisión, costos, tiempo y facilidad de implementación. Destaca investigaciones que demuestran la efectividad de los drones en la inspección de edificaciones y puentes. Este estudio es relevante para el proyecto de tesis, ya que proporciona información sobre el impacto de los drones en la construcción y su uso para analizar la integridad estructural y predecir fallas en puentes con modelos de DL.

Por otro lado, Maguiña y Pascacio, (2021) presentaron un trabajo sobre una herramienta digital para inspeccionar puentes. Usaron una red neuronal y visión por computadora para detectar fisuras y vegetación en puentes en Ancash, Perú. Proponen automatizar la detección para ahorrar tiempo y dinero. Crearán dos páginas web con Docker para probar las herramientas. Los autores mencionan que estas herramientas detectaron grietas y daños por vegetación en estructuras. Los resultados incluirán métricas de efectividad. El trabajo se conecta con la tesis, que usa aprendizaje profundo y visión por computadora para evaluar la salud de los puentes.

Palomino (2022) resalta la importancia de cuidar los monumentos coloniales, como el Puente Sutuc Chaca, por su valor histórico y porque sigue siendo usado como puente peatonal y vehicular. En el trabajo, se explora el uso de nuevas tecnologías, como drones y fotogrametría, para revisar y diagnosticar los daños estructurales de manera eficiente. Este enfoque se conecta con la tesis, que investiga cómo las tecnologías visuales, como los drones, pueden ayudar a evaluar la estructura de puentes históricos. Esto es relevante para la investigación sobre el uso de modelos con CNNs para supervisar puentes mediante imágenes.

Cañarte (2022) resalta la importancia de usar técnicas documentales e instrumentales para revisar puentes metálicos. Habla sobre la necesidad de identificar y entender los daños que pueden afectar su estructura, tanto a corto como a largo plazo. También menciona que es clave medir la gravedad de los daños, considerando el material y la vida útil del puente. Este enfoque se relaciona con la tesis al destacar la importancia de una inspección visual detallada para evaluar la estructura de los puentes. Es esencial para desarrollar modelos de aprendizaje profundo que predigan fallas a partir de la exploración visual de imágenes, lo que ayuda a aplicar evaluaciones precisas y sistemáticas.

De forma similar, Márquez y Restrepo (2022) describieron la importancia de las vías para el desarrollo económico. Subrayan la necesidad de mantener su integridad. Hicieron una inspección visual en Pereira, Colombia, analizando puentes y señalización. Usaron métodos del Instituto Nacional de Vías. Encontraron tramos dañados y grietas longitudinales por el tráfico. También estudiaron el deterioro de puentes y señalización por tráfico, vandalismo y clima. Este estudio se relaciona con la tesis, enfocándose en evaluar puentes con análisis visual.

1.3 JUSTIFICACIÓN

La infraestructura vial, en particular los puentes, son claves en la conectividad y el flujo seguro de personas y bienes. Sin embargo, garantizar la integridad estructural de los puentes puede ser complicado. Como se ha comentado, la inspección visual tradicional, aunque es una práctica común, puede ser limitada en su capacidad para detectar problemas estructurales o para proporcionar una evaluación a detalle de la salud estructural de los puentes. Ante este escenario, surge la necesidad de abordar estos desafíos con enfoques innovadores y tecnológicamente avanzados.

Para hacer más eficiente la inspección visual de puentes de concreto, este proyecto propone usar modelos de aprendizaje profundo, específicamente una CNN basada en VGG 16 implementada en Python. La idea es que la inspección visual de imágenes de puentes, junto con el análisis de datos y el DL, pueda ofrecer una evaluación más precisa y detallada de la condición de los puentes.

Este enfoque se justifica porque puede mejorar mucho la seguridad y la eficiencia de las carreteras y puentes. Al emplear técnicas avanzadas de análisis de datos, el modelo de CNN basado en la arquitectura VGG 16, se entrenará utilizando un conjunto diverso de imágenes de puentes de concreto y daños asociados. Esta amplia gama de datos permitirá al modelo identificar patrones significativos relacionados con el daño o el deterioro estructural, lo que a su vez facilitará la detección temprana de posibles fallas.

Además, la validación de este modelo mediante pruebas con imágenes de puentes de concreto no incluidas en el conjunto original garantizará su eficacia y capacidad de generalización en entornos diversos y situaciones reales. Los resultados obtenidos pueden ayudar a entender mejor cómo evaluar la condición de la infraestructura vial. También servirán como base para futuras investigaciones y para su uso práctico.

1.4 OBJETIVOS

1.4.1 Objetivo General

Evaluar la integridad estructural de puentes de concreto mediante el análisis de imágenes usando un modelo de aprendizaje profundo para detectar patologías o deterioros en los puentes.

1.4.2 Objetivos Específicos

- I. Efectuar una revisión de la literatura del trabajo relacionado al proyecto de tesis identificando fuentes relevantes y analizando artículos actuales para entender el contexto de la problemática propuesta.
- II. Implementar una arquitectura de Red Neuronal Convolucional (CNN) basada en VGG 16 adecuada usando Python para la detección de patrones relacionados con patologías y deterioro estructural en las imágenes de los puentes.

- III. Preparar el sistema de IA entrenando el modelo CNN basado en VGG16 usando las imágenes del *building-inspection-toolkit* para encontrar patrones de daños en los puentes.
- IV. Evaluar el modelo entrenado con un conjunto de imágenes de puentes de concreto no incluidas en el conjunto original, para validar su eficacia y capacidad de generalización en la detección temprana de problemas estructurales.

1.5 ALCANCES Y LIMITACIONES

1.5.1 Alcances

1. Desarrollo de un modelo de aprendizaje profundo: Implementación de una red neuronal convolucional utilizando Python y bibliotecas especializadas como TensorFlow y Keras para la evaluación de integridad estructural de puentes.
2. Preprocesamiento de Datos: Creación de algoritmos para preparar y procesar imágenes de puentes de concreto. Se incluyen técnicas como aumento de datos y normalización para hacer el modelo más robusto.
3. Clasificación de patologías: Entrenamiento del modelo para predecir posibles fallas en puentes mediante la identificación temprana de anomalías o deterioros estructurales.

1.5.2 Limitaciones

1. Disponibilidad de Datos: El modelo dependerá de la cantidad y calidad de las imágenes de puentes de concreto. Para que funcione bien en distintos casos, es importante que los datos sean variados y representativos.
2. Sensibilidad a Condiciones de Iluminación y Perspectiva: El modelo puede ser sensible a variaciones en las condiciones de iluminación y perspectiva en las imágenes de los puentes, lo que podría afectar su capacidad para detectar daños de manera precisa.
3. Limitaciones de Hardware y Tiempo de Procesamiento: El entrenamiento y la evaluación de modelos de Deep Learning pueden requerir recursos computacionales significativos, especialmente para conjuntos de datos

grandes. La disponibilidad de hardware adecuado puede ser una limitación en términos de tiempo y costos.

4. Interpretación de Resultados: Entender los resultados del modelo puede ser complicado y necesita conocimientos en ingeniería estructural para evaluarlos y tomar decisiones correctamente.

1.6 HIPÓTESIS

1.6.1 Hipótesis General

El uso de un modelo de aprendizaje profundo permite detectar de manera precisa las patologías o deterioros estructurales en puentes de concreto mediante el análisis de imágenes.

1.6.2 Hipótesis Específicas

1. Una revisión de la literatura existente sobre la evaluación estructural de puentes mediante aprendizaje profundo revelará enfoques y técnicas efectivas, proporcionando un contexto sólido para el desarrollo de un modelo CNN adaptado a la detección de fallas.
2. El diseño y desarrollo de una arquitectura basada en VGG 16 con ajustes adecuados permitirá detectar con precisión patrones relacionados con el deterioro estructural en las imágenes de los puentes.
3. El entrenamiento del modelo CNN con imágenes del conjunto de imágenes de inspección de edificios permitirá que el sistema aprenda a identificar patrones de daños en los puentes de concreto con un nivel de precisión aceptable.
4. El modelo entrenado será capaz de generalizar y detectar con eficacia problemas estructurales en imágenes nuevas de puentes no utilizadas durante el entrenamiento, demostrando su capacidad para realizar predicciones precisas en diferentes contextos y condiciones.

II. MARCO TEÓRICO

2.1 INTRODUCCIÓN TEÓRICA

Este capítulo explica la teoría relacionada con el proyecto. También muestra el contexto necesario para entender cómo se usa un modelo de Redes Neuronales Convolucionales en la evaluación de la integridad de puentes mediante imágenes. Este marco teórico se basa en cuatro áreas principales: 1. Ingeniería estructural, 2. Aprendizaje Profundo, 3. Procesamiento de imágenes, y 4. Sistemas de monitoreo de salud estructural en puentes (SHM, por sus siglas en inglés).

2.2 INGENIERÍA ESTRUCTURAL

La ingeniería estructural es una rama de la ingeniería civil. Se enfoca en diseñar y analizar estructuras para que sean estables, resistentes y duraderas (Jiménez, 2010). Los ingenieros estructurales trabajan en diversas construcciones, como edificios, puentes, presas y torres de comunicación.

Esta disciplina es clave para evaluar la integridad de los puentes (Monleón, 2017). No solo se ocupa del diseño y la construcción, sino también del mantenimiento y la seguridad a lo largo del tiempo. Evaluar la integridad estructural es fundamental para garantizar que los puentes sigan siendo seguros y funcionales. Esto previene fallos graves y reduce costos de mantenimiento.

El monitoreo de puentes de concreto es una tarea esencial dentro de la ingeniería estructural. Consiste en aplicar técnicas y tecnologías avanzadas para evaluar y mantener los puentes en buenas condiciones. Un seguimiento constante permite detectar daños a tiempo. Así, se pueden tomar medidas preventivas o correctivas antes de que los problemas sean más graves.

El uso de sensores y sistemas de monitoreo, como los sensores de fibra óptica y los sistemas de monitoreo dinámico, ayuda a los ingenieros a obtener información en tiempo real sobre cómo se comportan los puentes bajo diferentes cargas y condiciones ambientales. Estos datos son esenciales para analizar el estado del puente y detectar áreas que puedan requerir atención urgente.

Además, las inspecciones visuales complementan el monitoreo tecnológico. Los ingenieros realizan inspecciones regulares para revisar aspectos como fisuras, corrosión y el desgaste del concreto y de los refuerzos. La combinación de ambas técnicas, el monitoreo avanzado y la inspección visual, ofrece un enfoque completo para asegurar la seguridad y la durabilidad de los puentes de concreto.

La supervisión y evaluación constantes no solo mejoran la seguridad y el funcionamiento de los puentes, sino que también ayudan a reducir los costos de mantenimiento al permitir intervenciones a tiempo y evitar reparaciones costosas en el futuro. La ingeniería estructural, especialmente el monitoreo de puentes de concreto, es clave para mantener las infraestructuras importantes y proteger la seguridad de la comunidad. A continuación, se describen los diferentes tipos de puentes, sus problemas o daños comunes y sus características, seguido de una explicación del sistema de monitoreo como el propuesto en la presente tesis.

2.2.1 Puentes

Los puentes son estructuras diseñadas para permitir el paso sobre obstáculos naturales o artificiales, como ríos, valles, carreteras u otras vías de transporte. Su función primaria es ofrecer un camino estable y seguro para el tráfico humano y vehicular (Monleón, 2017; Núñez et al., s.f.). Existen varios tipos de puentes, cada uno con características específicas adaptadas a diferentes situaciones y necesidades. La Figura 2 ilustra los componentes clásicos de un puente genérico.

En general, según su propósito y el tipo de condiciones, los puentes pueden diseñarse como puentes de vigas, puentes de arco, puentes de armadura, puentes colgantes, puentes en voladizo, y puentes suspendidos en cables. La Figura 3, muestra un ejemplo de cada tipo de puente.

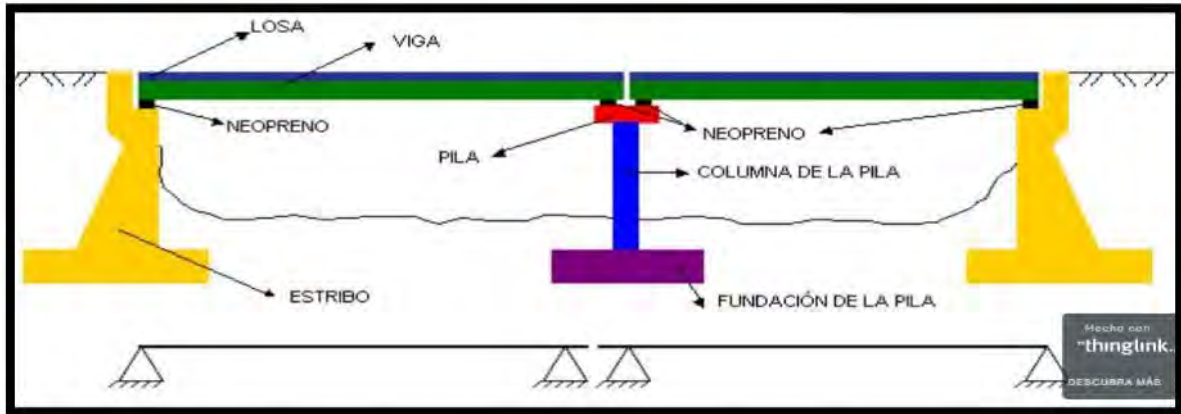


Figura 2. Componente de un puente. (INCAP).

Los puentes de vigas son estructuras simples y baratas. Están formados por vigas horizontales que se apoyan en pilares o soportes en sus extremos. Pueden ser de un solo tramo o de varios, lo que les permite cubrir distancias diferentes. Las vigas pueden ser de materiales como hormigón, madera o acero, según el diseño y las condiciones del lugar. Estos puentes se usan mucho en ciudades y campos porque son fáciles de construir y versátiles.

Los puentes arco se caracterizan por tener una estructura curva que distribuye las cargas hacia los extremos, generalmente pilares o estribos. Esta forma arqueada les permite resistir fuerzas laterales y cubrir grandes luces sin necesidad de soportes intermedios. Los puentes arco pueden ser de hormigón, acero o mampostería, y su diseño puede variar desde arcos simples hasta arcos múltiples y arcos segmentados.

Los puentes de armadura tienen una estructura de barras rectas unidas en triángulos para formar una armazón. Esta forma les da mucha resistencia y poco peso, lo que los hace perfectos para cubrir largas distancias. Se usan mucho en ferrocarriles y carreteras, donde se necesita una estructura fuerte y duradera.



Figura 3. Tipos de puentes. De izquierda a derecha, y de arriba hacia abajo. Puente Chivis, ubicado en el sector Puente Raither–Puerto Bermúdez, en la región Pasco. Puente de Piedra (Lima). Puente modular Choquehuanca. Puente colgante Perené. Puente en voladizo de Flandes. Puente suspendido en cables de Capachari.

Los puentes colgantes están sostenidos por cables o cadenas que se extienden sobre torres o pilares. La plataforma del puente se mantiene con tirantes verticales llamados suspendidos. Estos puentes son perfectos para cubrir largas distancias. Son conocidos por su estética y resistencia a las fuerzas de tensión. Algunos de los puentes colgantes más conocidos, como el Golden Gate en San Francisco y el puente

de Brooklyn en Nueva York, son símbolos de la ingeniería moderna y del diseño distintivo.

Los puentes voladizos se construyen en voladizo desde los pilares o estribos, extendiéndose hacia afuera desde un punto de apoyo central. Este tipo de puente se construye por secciones sucesivas que se van extendiendo gradualmente desde cada extremo hasta que se encuentran en el centro. Los puentes voladizos son útiles cuando se necesita mantener el tráfico fluvial o marítimo sin interrupciones significativas durante la construcción.

Los puentes suspendidos en cables son un tipo de puente colgante. Su plataforma se sostiene con cables que cuelgan de torres o pilares. Los cables están sujetos de forma vertical, lo que mantiene el puente elevado y estable. Estos puentes son ideales para cubrir grandes distancias y resistir fuerzas de tensión. Los cables principales soportan el peso de la plataforma, mientras que los cables de suspensión distribuyen la carga. Se usan en lugares donde se necesita mucha capacidad de carga y una vista impresionante, como en puentes largos que cruzan grandes cuerpos de agua.

2.2.2 Patologías en puentes

Uno de los aspectos más importantes al evaluar la integridad de los puentes es estudiar los daños y el desgaste que pueden sufrir. Esto incluye conocer los tipos de problemas más comunes que afectan estas estructuras (Enright y Frangopol, 2000; Bieñ et. al., 2020; Zhang y Yuen, 2022). Entre los daños más frecuentes se encuentran (Ver Tabla 1):

1. Las grietas.
2. La corrosión.
3. Barras de refuerzo expuestas.
4. Eflorescencia.
5. Descascaramiento.
 - a. Encalado (Scaling)
 - b. Desprendimiento (Spalling)

Las grietas en los puentes pueden ser causadas por varios factores, como el desgaste del material, cargas excesivas, terremotos o cambios de temperatura. Estas fisuras pueden ser pequeñas o grandes y van en diferentes direcciones. Su presencia muestra que el puente está soportando más presión de la que puede aguantar. Las grietas pueden debilitar el puente y, si no se reparan, pueden empeorar. Es importante encontrar y arreglar las grietas a tiempo para mantener el puente seguro y estable.

La corrosión es un problema importante que puede debilitar la estructura de los puentes con el tiempo, provocando fallos estructurales (Tinnea et al., 2006; Chen et al., 2021). El desgaste de los materiales, causado por el clima y el tráfico, también contribuye al deterioro de los puentes.

El óxido es especialmente problemático en puentes de acero. Se forma cuando el acero entra en contacto con agua y oxígeno, convirtiéndose en óxido de hierro. Este proceso debilita el acero, reduce su resistencia y, en casos graves, puede causar fallos estructurales. La presencia de óxido indica que la protección del puente, como la pintura, está fallando y necesita ser reparada para evitar daños mayores.

Las barras de refuerzo expuestas en estructuras de concreto son un claro signo de deterioro. Este daño puede ser consecuencia de la pérdida de cobertura de concreto debido a la corrosión, el agrietamiento, o el impacto mecánico. Cuando las barras de refuerzo quedan expuestas, están más susceptibles a la corrosión, lo que puede debilitar la estructura y comprometer su capacidad portante. La reparación de este tipo de daño suele implicar la restauración del recubrimiento de concreto y la protección de las barras de refuerzo.

La eflorescencia es la aparición de sales cristalinas en la superficie del concreto (Uchasara, 2022). Sucede cuando el agua disuelta en las sales migra a través del material y se evapora, dejando depósitos blancos. Aunque la eflorescencia no daña la estructura, puede indicar problemas de humedad que, a largo plazo, afectan la durabilidad del concreto y favorecen la corrosión del acero.

Tabla 1. Principales tipos de daños en puentes estudiados en esta tesis.

Daño	Descripción
Sin Daño	La etiqueta "Sin Daño" describe imágenes que contienen una superficie de concreto saludable o contenido irrelevante.
Grieta	Las grietas aparecen cuando se excede la resistencia a la tracción del concreto o durante el endurecimiento, si el post-tratamiento o la receta del concreto eran inadecuados.
Eflorescencia	La eflorescencia es generalmente blanquecina o amarillenta. Aparece cuando se disuelven sales (calcio, sodio, potasio) de la piedra de cemento. La piedra de cemento es el material duro que se forma cuando el cemento y el agua se mezclan. Esta pasta une la arena y la grava, que son los otros componentes del concreto. Este es un defecto frecuente que surge cuando la parte correspondiente de la construcción está en contacto constante con agua corriente.
Descascaramiento	El descascaramiento puede aparecer debido a cambios de congelación-deshielo, corrosión del refuerzo subyacente o impacto, por ejemplo, de autos que golpean la estructura. Puede incluir encalado (scaling) o desprendimiento (spalling).
Óxido	El óxido aparece en objetos metálicos como el refuerzo y el concreto. El óxido visible en la superficie del concreto se origina en partes metálicas vecinas o en el refuerzo subyacente. El refuerzo puede corroerse como resultado de la pérdida de la capa protectora alcalina proporcionada por el concreto no carbonatado. Si el valor de pH baja debido a la carbonatación del concreto, que es inevitable con el tiempo, el refuerzo puede oxidarse.
Barras Expuestas	Refuerzo visible debido al descascaramiento del concreto superpuesto, que generalmente es consecuencia de la corrosión del refuerzo. Otra causa de refuerzo visible son los "bolsillos de roca" que surgen si la pasta de cemento no llenó todo el volumen entre el agregado grueso. Los "bolsillos de roca" pueden formarse si las propiedades reológicas del concreto o su compactación fueron inadecuadas.

También es importante tener en cuenta el impacto del descascaramiento en la estructura del concreto. El descascaramiento, o desprendimiento superficial, puede debilitar el material, exponiendo el refuerzo a agentes corrosivos y reduciendo su protección. Este fenómeno suele ser causado por ciclos de congelación y deshielo, corrosión del refuerzo o impactos mecánicos, como los de los vehículos que golpean la estructura.

El descascaramiento no solo afecta la apariencia estética del concreto, sino que también puede acelerar el deterioro del refuerzo, lo que a su vez puede llevar a fallos estructurales si no se aborda adecuadamente. Por lo tanto, es crucial realizar inspecciones periódicas y tomar medidas preventivas para mitigar estos efectos y asegurar la longevidad de las estructuras de concreto.

El descascaramiento puede presentarse en dos formas: escalado y desprendimiento. El escalado es un defecto del concreto en el que la superficie se desprende o se desintegra en capas finas. Esto generalmente ocurre por la exposición al agua y los ciclos de congelación/descongelación, o por una mala calidad del concreto en la superficie.

Por otro lado, el desprendimiento es un defecto donde se pierden pequeñas áreas de la superficie. Suele ocurrir por la corrosión de las armaduras o daño físico. Puede ser causado por estrés mecánico, clima extremo o productos químicos.



Figura 4. Daños comunes en puentes. De las patologías en puentes, en este proyecto, se analizan estos tipos de daños: Grietas, Eflorescencia, y Descascaramiento (Flotzinger, J., et al., 2023).

Cada tipo de daño necesita un enfoque específico para identificarlo y repararlo (Garay y Trejo, 2021; Flotzinger et al., 2023). La Tabla 1 muestra estos tipos de daños. Los ingenieros estructurales usan técnicas como inspección visual, pruebas no invasivas

y/o monitoreo continuo para evaluar el estado de los puentes y planear las reparaciones necesarias. Entender y tratar bien estos daños es fundamental para asegurar la seguridad y la duración de los puentes. La Figura 4 presenta los tipos de daños estudiados en esta tesis. A continuación, se explican estas patologías según su naturaleza y causas.

2.2.3 Origen de las patologías en puentes

La naturaleza de las patologías, según S&P (2019), presenta una clasificación en función de sus agentes causantes, es decir, dependiendo del origen en que se ha dado. Por ello, se han determinado las lesiones físicas, mecánicas y químicas.

Lesiones Físicas. Este tipo de daño es causado por la interacción de fuerzas físicas que afectan las estructuras. Entre los más frecuentes se encuentran la humedad, la acumulación de suciedad y la erosión.

1. Lesiones Físicas. Este tipo de daño es causado por la interacción de fuerzas físicas que afectan las estructuras. Entre los más frecuentes se encuentran la humedad, la acumulación de suciedad y la erosión.
2. Humedad: Es una de las complicaciones más comunes y graves en la construcción, ya que afecta directamente la calidad del ambiente y la habitabilidad. Las principales causas incluyen filtraciones, tuberías rotas, ascenso capilar del agua y la presencia de puentes térmicos.
3. Suciedad: Los materiales porosos, que suelen ser de origen natural y tienen la capacidad de absorber líquidos y gases, actúan de manera similar a una esponja. Cuando estos materiales están expuestos a la contaminación ambiental, favorecen la acumulación de suciedad, especialmente en las fachadas.
4. Erosión: Este fenómeno es causado principalmente por factores climáticos. Cuando el agua se filtra por las aberturas de la estructura y experimenta una disminución considerable de temperatura, se congela, causando una

expansión del volumen y, como resultado, el deterioro del material por medio de grietas.

Lesiones mecánicas, las cuales surgen debido a una sobrecarga en los elementos de la estructura o por fuerzas externas o internas. Estas afectan tanto el aspecto estructural como el constructivo y su uso.

1. Grietas y fisuras: Son rupturas en los elementos de la estructura. Las fisuras son superficiales, mientras que las grietas afectan más profundamente el componente constructivo.
2. Desprendimientos: Se producen cuando se pierde la conexión entre los elementos de la estructura, por diversas causas que pueden ser identificadas.
3. Erosión: Se origina por la fricción o el viento, afectando principalmente los bordes de los puentes.
4. Deformaciones: Son alteraciones en la forma original de los elementos estructurales, causadas por esfuerzos o cargas prolongadas que modifican su estructura.

Por último, se presentan las lesiones químicas, que son causadas por reacciones químicas en los materiales, la acción de factores atmosféricos, la contaminación y la actividad de organismos vivos. Algunas de estas lesiones son:

1. Eflorescencias: Son cristales blancos de sales que aparecen específicamente en superficies de concreto. Las sales se mueven por materiales porosos y se muestran cuando el agua se evapora por el sol.

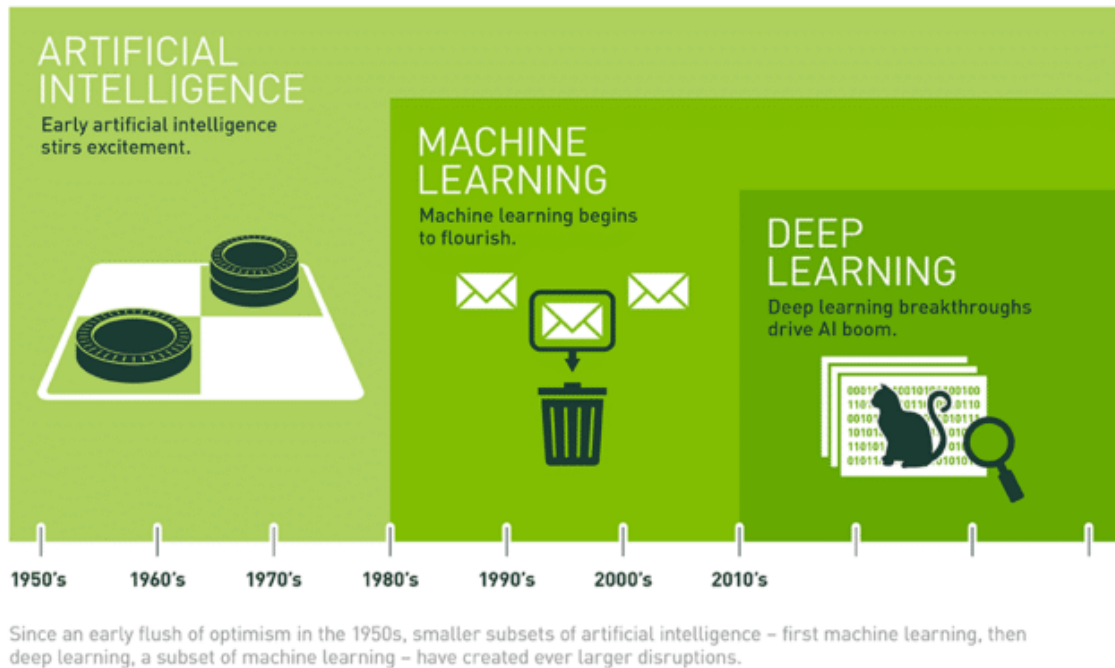


Figura 5. Tipos de inteligencia artificial. El aprendizaje profundo forma parte del aprendizaje de máquina que a su vez está incluido en la inteligencia artificial. (Copeland, M. 2021).

2. Oxidación: Sucede cuando los metales se dañan. Esto puede causar que aumenten de volumen o se disuelvan, especialmente cuando el óxido permanece húmedo.
3. Corrosión: Es la pérdida del material metálico debido a reacciones químicas entre el metal y otros materiales cercanos. Este proceso, causado por la oxidación, afecta a todos los metales.

2.3 APRENDIZAJE DE MÁQUINA

El aprendizaje de máquina (AM) es un subcampo de la inteligencia artificial como se muestra en la Figura 5. El AM está cambiando rápido. Se enfoca en crear sistemas que aprenden de los datos. Pueden hacer predicciones o tomar decisiones sin necesidad de estar programados para cada tarea. (Jiménez, 2021). A diferencia de la

programación tradicional basada en reglas, cada paso es predeterminado por el programador. Los algoritmos de AM permiten a las computadoras identificar patrones. También pueden obtener conocimientos y mejorar su rendimiento. Esto ocurre de manera autónoma mientras procesan más datos. Esta capacidad de adaptación convierte al AM en una herramienta importante de la IA, permitiendo que los sistemas evolucionen en función de la experiencia y perfeccionen sus resultados con mayor precisión.

En el núcleo del AM está el concepto de usar datos para entrenar algoritmos. Estos algoritmos operan analizando grandes conjuntos de datos para detectar patrones, relaciones y tendencias ocultas que pueden no ser evidentes para los humanos. Una vez entrenado, el sistema puede generalizar a partir de experiencias pasadas para tomar decisiones informadas o hacer predicciones cuando se enfrenta a nuevos datos desconocidos. Por ejemplo, el AM es el motor detrás de los motores de recomendación utilizados por minoristas en línea, los sistemas de reconocimiento de voz en asistentes virtuales y los complejos procesos de toma de decisiones en vehículos autónomos.

2.3.1 Aprendizaje supervisado

El AM tiene varias categorías, cada una para tareas específicas. El aprendizaje supervisado es una de ellas. Consiste en entrenar un modelo con un conjunto de datos etiquetados, donde se tienen los datos de entrada y sus resultados esperados (Valenzuela, 2022). Esto permite que el algoritmo aprenda de ejemplos, y se usa mucho para tareas como clasificación y regresión.

En el aprendizaje supervisado, el modelo se entrena con un conjunto de datos que incluye entradas etiquetadas y sus salidas correspondientes. Así, el modelo aprende cómo se relacionan los datos de entrada con los resultados esperados. Luego, usa este conocimiento para hacer predicciones o tomar decisiones con datos nuevos.

El aprendizaje supervisado se usa en muchas áreas. Por ejemplo, en tareas de clasificación, el objetivo es asignar una etiqueta o categoría a una entrada según características conocidas. Un ejemplo común es clasificar correos electrónicos como “spam” o “no spam”. El modelo se entrena con ejemplos etiquetados de correos spam

y no spam, aprendiendo a diferenciarlos según características como el contenido y el remitente.

La regresión es otro tipo de tarea en el aprendizaje supervisado, donde el objetivo es predecir un valor continuo en lugar de una categoría discreta. Un ejemplo de regresión es la predicción del precio de una vivienda basado en características como el tamaño, la ubicación y el número de habitaciones. En este caso, el modelo se entrena con datos históricos de viviendas, donde se conocen tanto las características de la vivienda como su precio de venta. Luego, el modelo utiliza esta información para estimar el precio de nuevas viviendas basándose en sus características.

El entrenamiento en el aprendizaje supervisado consiste en ajustar los parámetros del modelo para reducir la diferencia entre las predicciones del modelo y los resultados reales en los datos de entrenamiento. Este ajuste se hace con algoritmos que buscan mejorar la precisión del modelo en la tarea para la que fue entrenado. Después de entrenar y validar el modelo, se puede usar con nuevos datos para hacer predicciones o clasificaciones, basándose en lo que aprendió durante el entrenamiento.

2.3.2 Aprendizaje no supervisado

El aprendizaje no supervisado trabaja con datos que no tienen etiquetas. El objetivo es que el algoritmo encuentre patrones o grupos en los datos sin saber exactamente qué buscar (Rojas, 2020). Se usa mucho en segmentación de mercado, detección de problemas y reducción de la cantidad de datos. En este tipo de aprendizaje, uno de los objetivos principales es la agrupación o *clustering*. El modelo organiza los datos en grupos o *clusters*, donde los datos dentro de cada grupo son más parecidos entre sí que con los de otros grupos. Un ejemplo es agrupar a los clientes en marketing según sus hábitos de compra. Esto ayuda a las empresas a crear campañas más efectivas para cada grupo.

Otro objetivo importante en el aprendizaje no supervisado es la reducción de la dimensionalidad. Este proceso implica simplificar los datos al reducir el número de variables o características, manteniendo al mismo tiempo la mayor cantidad posible de información relevante. Técnicas como el Análisis de Componentes Principales (PCA) son utilizadas para transformar datos de alta dimensión en una forma más

manejable y comprensible. Esta reducción es útil para visualizar datos complejos y para mejorar la eficiencia de otros algoritmos de aprendizaje automático al eliminar el ruido y las redundancias en los datos.

El aprendizaje no supervisado también incluye la detección de anomalías, donde el objetivo es identificar datos que se desvían significativamente del patrón general. Esta técnica es valiosa en aplicaciones como la detección de fraudes en transacciones financieras o la identificación de fallos en sistemas de mantenimiento predictivo. Al detectar anomalías, se pueden tomar medidas correctivas antes de que los problemas se conviertan en situaciones críticas.

2.3.3 Aprendizaje por refuerzo

El aprendizaje por refuerzo es diferente. En este caso, un agente aprende a tomar decisiones interactuando con su entorno. Recibe retroalimentación en forma de recompensas o castigos (Güitta, 2024). Este enfoque es útil en aplicaciones como juegos, robótica y sistemas que requieren decisiones complejas.

En el aprendizaje por refuerzo, un agente explora un entorno donde puede hacer varias acciones. Cada acción cambia el entorno y afecta la recompensa que recibe. El agente usa esta información para mejorar su forma de actuar. Lo más importante es el balance entre explorar cosas nuevas y usar las estrategias que ya sabe que funcionan.

Un concepto clave en el aprendizaje por refuerzo es la función de valor, que predice la recompensa futura de cada acción en un estado. Los algoritmos, como el Q-learning o las redes neuronales, actualizan estas predicciones usando la experiencia acumulada. La política del agente, que es su estrategia para elegir acciones, se ajusta constantemente para conseguir la mayor recompensa posible.

Un ejemplo común de aprendizaje por refuerzo es entrenar agentes para jugar videojuegos. El agente juega y aprende de la retroalimentación del juego, mejorando su puntuación o superando niveles. Esto también se usa en robótica, donde los robots aprenden a hacer tareas difíciles, como mover objetos o explorar lugares nuevos, ajustando sus acciones para alcanzar metas específicas.

El aprendizaje por refuerzo es útil cuando el entorno cambia y las consecuencias de las acciones no se ven de inmediato. Con la práctica y la retroalimentación, el agente aprende a tomar decisiones que le dan mejores resultados a largo plazo, no solo recompensas rápidas. Esto lo hace útil para problemas complejos, como mejorar procesos industriales o para coches autónomos.

2.3.4 Aprendizaje profundo

El Aprendizaje Profundo (DL) es una parte del AM que enseña a los modelos de inteligencia artificial a realizar tareas por sí mismos. Utiliza algoritmos basados en cómo funciona el cerebro humano (Dong et al., 2021).

Este método es muy útil para trabajar con datos complejos, como imágenes, sonidos y textos (Chai et al., 2021; Minaee et al., 2021; Jovanovic et al., 2024). Ha mejorado cosas como el reconocimiento de objetos, la traducción de idiomas y la conducción autónoma. En el caso de los puentes, el aprendizaje profundo ayuda a analizar imágenes para detectar daños o problemas de forma precisa y rápida.

En particular, el DL se basa en Redes Neuronales Artificiales (ANN) con múltiples capas (de ahí el término "profundo") (Muthukrishnan et. al., 2020), que pueden aprender representaciones de datos cada vez más abstractas y complejas a medida que se profundiza en la red. Las ANNs son estructuras computacionales inspiradas en el funcionamiento de las neuronas biológicas. Consisten en capas de nodos interconectados que procesan y transmiten información.

En una ANN, las capas ocultas están entre la capa de entrada y la de salida en una red neuronal. Cada capa oculta aprende representaciones más abstractas de los datos. Una red neuronal usada para clasificar imágenes es la Red Neuronal Convolutiva (CNN, por sus siglas en inglés) (Shahriar, 2023). La Figura 5 muestra la arquitectura de una CNN. Esta red está diseñada para trabajar con datos en forma de cuadrícula, como imágenes. Su diseño se inspira en cómo el

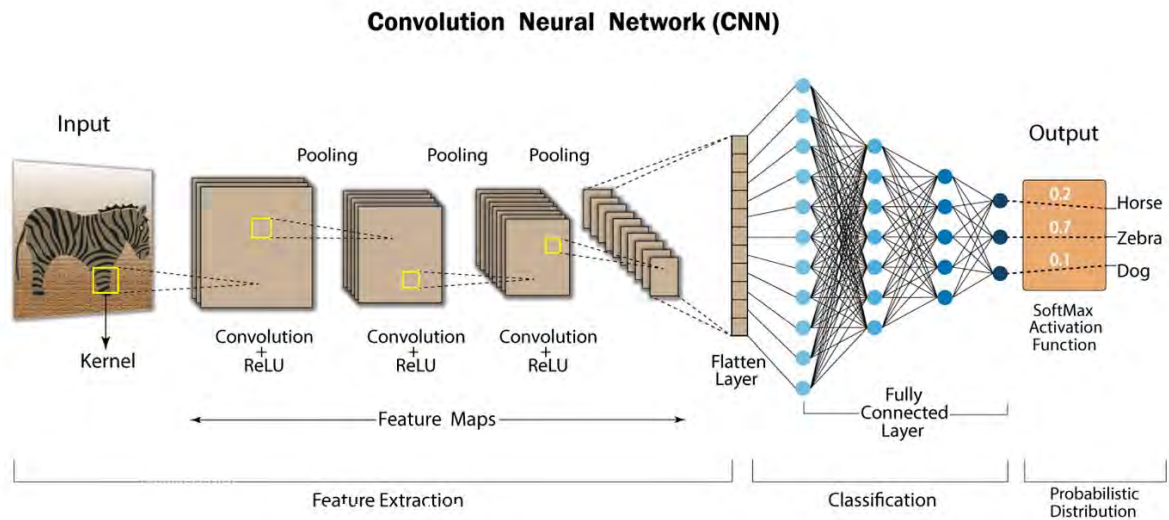


Figura 6. Arquitectura de una Red Neuronal Convolutiva (CNN). Las CNNs se componen de diferentes capas de redes neuronales con operaciones como la convolución (Convolution), Agrupación (Pooling). La profundidad de las CNNs les permite capturar características descriptivas de las imágenes (Shahriar, N. 2023).

cerebro animal organiza las neuronas, que responden sólo a estímulos en áreas específicas del campo visual.

Las CNN se caracterizan por sus capas convolucionales. Estas capas usan filtros que se aplican a los datos de entrada (Chen et al., 2021). Los filtros se mueven sobre la imagen y calculan productos escalares entre los píxeles de la imagen y los pesos del filtro. Esto ayuda a la red a aprender características locales como bordes y texturas en las primeras capas, y características más complejas en las capas posteriores.

Además de las capas convolucionales, las CNN suelen tener capas de agrupación (pooling), que reducen el tamaño de los datos. También incluyen capas completamente conectadas al final, que se encargan de hacer la clasificación o regresión final.

Las CNN han sido muy exitosas en tareas de visión computacional, como el reconocimiento de objetos, la detección de rostros y la segmentación de imágenes. También se usan en áreas como el procesamiento de señales de audio y la bioinformática. Su capacidad para aprender características automáticamente y su resistencia a cambios locales las hacen esenciales en el aprendizaje profundo.

2.4 PROCESAMIENTO DIGITAL DE IMÁGENES

El Sistema de Visión por Computadora es un campo interdisciplinario que se enfoca en permitir a las máquinas adquirir, procesar y analizar imágenes digitales para realizar tareas específicas (Dougherty, 2020). Utiliza algoritmos y técnicas de procesamiento de imágenes para interpretar y comprender la información visual contenida en las imágenes. Los sistemas de visión computacional para clasificación avanzada usan CNNs.

Los elementos principales de un sistema de visión por computadora son:

1. Captura de imágenes: Obtener imágenes con cámaras o sensores.
2. Procesamiento previo: Mejorar la calidad de las imágenes, como ajustar colores y quitar ruidos.
3. Segmentación: Dividir la imagen en partes importantes para facilitar su análisis.
4. Extracción de características: Identificar elementos clave como bordes o formas en la imagen.
5. Reconocimiento de patrones: Usar la información visual procesada para reconocer objetos o detectar problemas.

Este sistema es clave para analizar imágenes de manera avanzada y se usa en muchos campos, como la revisión de puentes con modelos de CNNs (Flotzinger et al., 2023). Para lograrlo, el sistema puede contar con lo siguiente:

1. Adquisición de Imágenes: Las cámaras de alta resolución y los drones se utilizan para capturar imágenes detalladas de la superficie del puente desde diferentes ángulos y posiciones. Los sensores especializados también pueden capturar datos en espectros que el ojo humano no puede ver, como el infrarrojo o el ultravioleta.
2. Preprocesamiento de Imágenes: Las imágenes capturadas pueden estar afectadas por variaciones de iluminación, ruido y otras distorsiones. Las técnicas de corrección de color, filtrado de ruido y mejora del contraste mejoran la calidad de las imágenes, facilitando el análisis posterior.

3. Segmentación de Imágenes: El software de visión por computadora segmenta las imágenes en diferentes regiones para identificar áreas de interés, como fisuras, corrosión y otras anomalías en la superficie del puente. Esta segmentación facilita el análisis específico de cada área.
4. Extracción de Características: Se utilizan algoritmos para extraer características relevantes de las imágenes segmentadas. Estas características pueden incluir la longitud y anchura de las fisuras, la textura de la superficie del concreto, y la forma y distribución de las manchas de corrosión.
5. Reconocimiento de Patrones: Las CNNs se entrenan con grandes conjuntos de datos de imágenes de puentes de concreto en buen estado y en mal estado. Estas CNNs pueden aprender a reconocer patrones específicos asociados con diferentes tipos de daños estructurales.

Una vez entrenadas, las CNNs analizan las imágenes procesadas para identificar y clasificar automáticamente los daños. Pueden detectar anomalías y patrones de deterioro que podrían ser indicativos de problemas estructurales serios.

En la práctica, estos sistemas hacen que las inspecciones de puentes sean más rápidas y precisas que los métodos tradicionales. Las inspecciones hechas por personas pueden ser imprecisas y cometer errores, sobre todo en condiciones difíciles o peligrosas. Los sistemas de visión por computadora, en cambio, dan una evaluación objetiva usando datos medibles. Además, los drones con cámaras de alta calidad y sensores especiales pueden inspeccionar lugares difíciles de alcanzar sin necesidad de andamios o equipos para subir, lo que reduce costos y riesgos.

La detección temprana de daños estructurales permite realizar reparaciones antes de que los problemas se agraven, mejorando así la seguridad del puente. De la misma forma, las intervenciones de mantenimiento pueden planificarse de manera más efectiva, optimizando los recursos y reduciendo el tiempo de inactividad del puente. También, al identificar problemas antes de que se conviertan en fallos catastróficos, se evitan reparaciones costosas y se prolonga la vida útil del puente.

2.5 SISTEMAS DE MONITOREO DE SALUD ESTRUCTURAL EN PUENTES

Los sistemas de monitoreo de la salud estructural (BHMS, por sus siglas en inglés) son herramientas avanzadas para revisar y garantizar la seguridad de los puentes (He et al., 2022). Usan sensores, análisis de datos y diagnósticos para encontrar, evaluar y predecir daños en las estructuras de los puentes. Un ejemplo de esto es el sistema de procesamiento de imágenes, que se usa para clasificar daños en los puentes analizando imágenes. Un BHMS, por lo general, tiene sensores distribuidos por todo el puente. Estos sensores miden cosas como deformaciones, vibraciones, temperatura, humedad y corrosión. Los sensores más comunes son acelerómetros, extensómetros, fibra óptica y sensores de corrosión.

Los acelerómetros son importantes para medir las vibraciones causadas por el tráfico y los terremotos. Proporcionan datos sobre cómo responde el puente. Los extensómetros miden las deformaciones en los materiales del puente. Esto ayuda a detectar cambios en su estructura. Los sensores de fibra óptica son muy precisos y duraderos. Se usan para monitorear deformaciones y temperaturas en grandes áreas del puente. Los sensores de corrosión detectan la presencia de corrosión en las armaduras de acero, alertando sobre la necesidad de intervenciones preventivas.

Los sensores envían la información a un sistema central. Allí se analiza en tiempo real usando algoritmos avanzados. Estos algoritmos comparan los datos actuales con modelos esperados y datos históricos. Cualquier desviación puede indicar daños o deterioro, activando alertas para los ingenieros.

El uso de IA en el análisis de datos mejora la precisión en la detección de anomalías. También ayuda a prever posibles fallos antes de que ocurran. Estos algoritmos identifican patrones complejos y correlaciones que no se verían con el análisis tradicional, proporcionando una evaluación más precisa y rápida de la salud del puente.

Un BHMS tiene muchos beneficios. Permite monitorear el puente de manera continua y en tiempo real. Esto ayuda a detectar problemas pronto y evitar daños graves. Así, se mejora la seguridad y se reducen los costos de mantenimiento, ya que se evitan reparaciones grandes. Además, el sistema guarda información detallada sobre el

comportamiento del puente con el tiempo. Esta data es muy útil para los ingenieros y las autoridades, ayudándoles a tomar decisiones basadas en hechos.

Particularmente, este proyecto de tesis se centra en la detección de daños en puentes usando información visual, es decir, imágenes centradas en un daño potencial para su clasificación. Los sistemas de visión pueden formar parte integral de los BHMS (Dong y Catbas, 2021). A continuación, el siguiente capítulo describe la metodología seguida para la ejecución de este proyecto de tesis.



III. METODOLOGÍA

En esta sección, se expone la metodología seguida en esta tesis para el desarrollo del sistema de clasificación de patologías en puentes propuesto. Esta metodología es mostrada en la Figura 6. Primero se explica el conjunto de imágenes de las patologías de puentes, y su preparación. Después, se detalla la forma de entrenamiento y evaluación del modelo de CNNs. Por último, se explica cómo este sistema puede ser explotado en un BHMS.

3.1 RECOLECCIÓN DE DATOS

3.1.1 Recopilación de datos

El conjunto de datos usando en este proyecto de tesis está disponible públicamente, y es descrito por Flotzinger, J., et. al. (2023). El conjunto de datos incluye imágenes de daños encontrados durante una inspección manual de puentes de concreto. Dentro de las patologías mostradas en el conjunto de datos están: Grietas, Eflorescencia, y Descascaramiento en sus dos niveles: Escalado y Desprendimiento. Algunas Imágenes de muestra se exponen en la Figura 7.

El conjunto de datos contiene 2,612 imágenes de diferentes puentes, y contienen ejemplos de las cuatro patologías ya mencionadas, así como ejemplos de imágenes de secciones saludables de puentes. Este conjunto de imágenes debe de ser preprocesado para mejorar la clasificación o identificación de patologías. Este proceso de preparación y limpieza se describe en la siguiente sección.

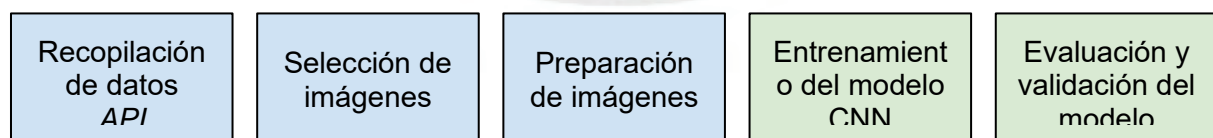


Figura 7. Metodología para el desarrollo del sistema de clasificación de patologías en puentes. Los primeros pasos (en azul) corresponden a los datos, mientras que los últimos dos pasos (en verde) corresponden al modelo CNN.

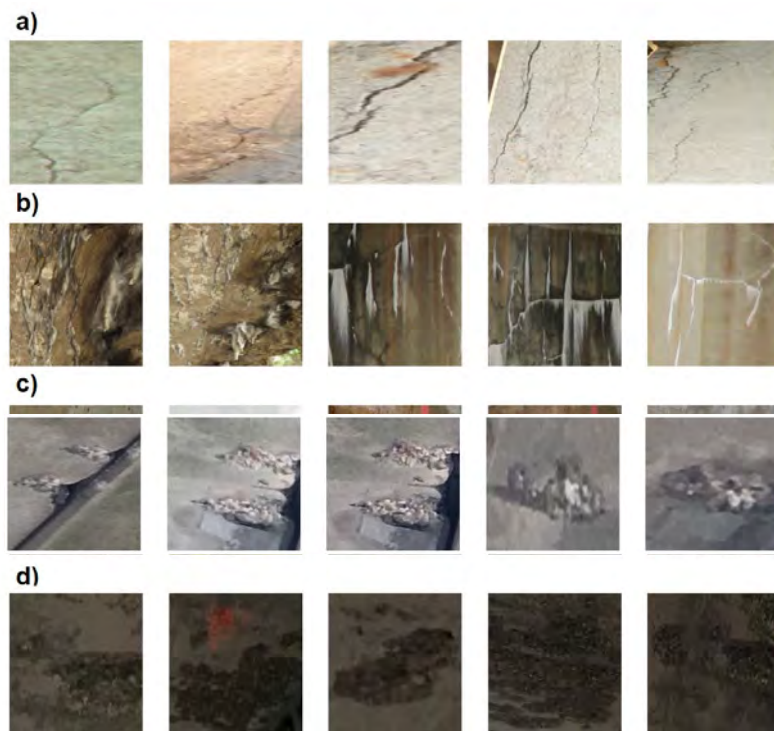


Figura 8. Ejemplos de imágenes de inspección de puentes de concreto. Las patologías analizadas en esta tesis son: a) Grietas, b) Eflorescencia, c) Escalado, y d) Desprendimiento.

3.2 PREPARACIÓN Y LIMPIEZA DE DATOS

La red neuronal CNN necesita que las imágenes de entrada sean tratadas para mejorar el entrenamiento. Para preparar imágenes es fundamental realizar un proceso de preprocesamiento que garantice la calidad y coherencia de los datos. Esto incluye varias etapas que son descritas a continuación.

1. Redimensionamiento: El redimensionamiento implica ajustar todas las imágenes a un tamaño uniforme, lo cual es necesario para que la CNN pueda procesarlas eficientemente. Esto también ayuda a reducir la carga computacional.
2. Normalización: Para la normalización hay que asegurarse de que los valores de los píxeles estén en una escala similar. Esto puede implicar escalar los valores de píxeles a un rango entre 0 y 1.

3. Aumento de datos: consiste en crear nuevas imágenes aplicando cambios aleatorios a las imágenes originales, como rotaciones, cambios de brillo o saturación. Esto ayuda a que el modelo sea más preciso y evite el sobreajuste.
4. División de entrenamiento y validación: se separa el conjunto de datos en dos partes: una para entrenar el modelo y otra para evaluar su rendimiento.
5. Codificación de etiquetas: si las etiquetas no están en formato numérico, se deben convertir a números para que el modelo las entienda.

3.3 ENTRENAMIENTO Y EVALUACIÓN DEL MODELO

3.3.1 Entrenamiento de la CNN

La red neuronal utilizada se llama VGG 16 (Mascarenhas y Agarwal 2021). Es muy buena para clasificar imágenes. "VGG" viene del grupo de investigadores de la Universidad de Oxford que la crearon. Esta red tiene capas convolucionales y capas conectadas. Las capas convolucionales buscan detalles importantes de las imágenes, como bordes y texturas. Las capas conectadas hacen la clasificación final.

VGG 16 tiene 16 capas en total. Estas están organizadas en bloques. Cada bloque tiene varias capas convolucionales seguidas de una capa de max pooling, que reduce la cantidad de información. Para este trabajo, se usó una versión preentrenada de VGG 16. Esto significa que los pesos de las capas ya han sido ajustados con datos grandes y variados. Así, la red puede usar características aprendidas de bases de datos como ImageNet (Morid et. al., 2021).

La última capa de la VGG 16 es una capa conectada que da las predicciones de las clases. Esta capa tiene tantas neuronas como clases hay en el problema. VGG 16 ha mostrado ser efectiva para clasificar muchos tipos de imágenes. Su estructura simple pero poderosa la hace popular para muchas aplicaciones de visión por computadora.

La VGG 16 es una red neuronal profunda y eficaz que se usa para clasificar imágenes. Su diseño en bloques y el uso de convoluciones y pooling le permiten extraer y aprender detalles importantes de las imágenes. Además, al usar pesos preentrenados, es muy útil cuando los datos para entrenar son limitados.

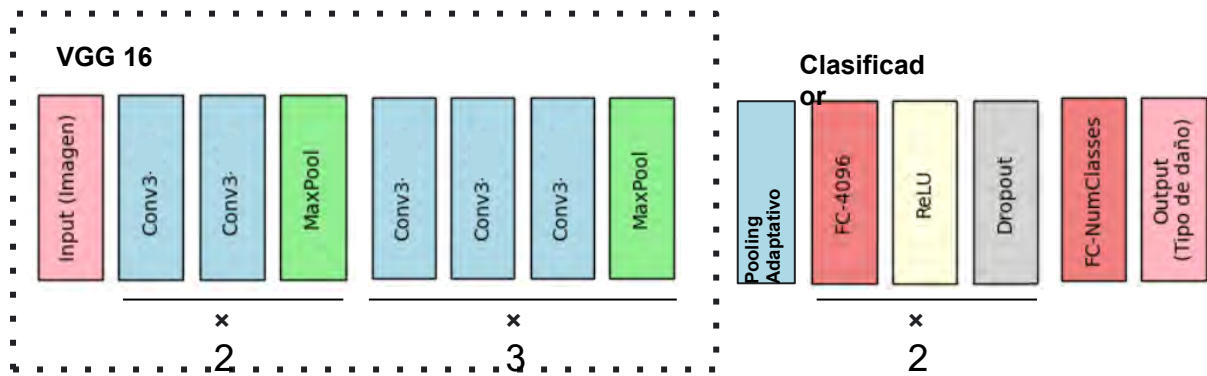


Figura 9. Arquitectura de CNN usada para la clasificación de daños en puentes. En este proyecto se adapta la arquitectura VGG 16 para la clasificación de patologías en puentes. Las capas adicionales mejoran la capacidad de la red para aprender características específicas del nuevo conjunto de datos mientras se reduce el riesgo de sobreajuste.

3.3.2 Descripción de la CNN

El modelo utilizado es adecuado para tareas de clasificación de imágenes, aprovechando las capas preentrenadas del VGG 16 original y añadiendo capas para adaptar la VGG 16 al problema de detección de patologías en puentes. Esta arquitectura adaptada permite beneficiarse de características previamente aprendidas en un conjunto de datos grande y diverso. La Figura 9 ilustra la arquitectura utilizada. Un resumen de la arquitectura es el siguiente:

Bloques VGG 16 (Preentrenado):

- Bloque 1: 2 capas convolucionales (64 filtros) + 1 capa de agrupamiento.
- Bloque 2: 2 capas convolucionales (128 filtros) + 1 capa de agrupamiento.
- Bloque 3: 3 capas convolucionales (256 filtros) + 1 capa de agrupamiento.
- Bloque 4: 3 capas convolucionales (512 filtros) + 1 capa de agrupamiento.
- Bloque 5: 3 capas convolucionales (512 filtros) + 1 capa de agrupamiento.

Estos bloques preentrenados se caracterizan por lo siguiente:

- Utilizan las capas convolucionales preentrenadas del modelo VGG-16 original.
- Están compuestos por una secuencia de capas convolucionales y de pooling.
- Cada conjunto de capas convolucionales está seguido por una capa de max pooling que reduce la dimensionalidad.
- Las capas convolucionales son responsables de extraer características de bajo y alto nivel de las imágenes.

Capas Adicionales (Adaptación):

Para conectar las capas de la VGG 16 al problema específico se añade una capa de pooling adaptativo y un bloque clasificador compuesto de varias capas.

1. Capa de pooling adaptativo:

- `nn.AdaptiveAvgPool2d((7, 7))` Ajusta automáticamente el tamaño de salida a 7x7, independientemente del tamaño de la entrada.
- Reduce aún más la dimensionalidad de las características extraídas antes de pasarlas a la red completamente conectada.

2. Clasificador:

- Primera capa totalmente conectada:
 - `nn.Linear(512 * 7 * 7, 4096)`
 - Conecta las características planas (a partir de la salida de la capa de pooling) a una capa de 4096 neuronas.
 - Función de activación: `nn.ReLU(True)`
 - Introduce no linealidad y permite que la red aprenda relaciones complejas.
 - Capa de Dropout: `nn.Dropout()`
 - Ayuda a prevenir el sobreajuste apagando aleatoriamente neuronas durante el entrenamiento.
- Segunda capa totalmente conectada:
 - `nn.Linear(4096, 4096)`

- Otra capa de 4096 neuronas para aumentar la capacidad de aprendizaje del modelo.
- Función de activación: `nn.ReLU(True)`
- Añade no linealidad adicional.
- Capa de Dropout: `nn.Dropout()`
- Apaga neuronas aleatoriamente para prevenir el sobreajuste.
- Tercera capa totalmente conectada:
 - ❖ `nn.Linear(4096, num_classes)`
 - ❖ Capa de salida que produce una predicción para cada una de las clases del conjunto de datos.
 - ❖ La cantidad de neuronas en esta capa es igual al número de clases (`num_classes`).

Adicionalmente, para el entrenamiento del modelo, se sigue el siguiente método:

- Paso de entrada: La imagen de entrada pasa primero a través de la capa de características (`self.features`).
- Paso de pooling: La salida de las características es procesada por la capa de pooling adaptativo (`self.avgpool`).

Tabla 2. Parámetros usados en el entrenamiento de la CNN.

Parámetro	Valor
Función de Pérdida	<code>nn.CrossEntropyLoss()</code>
Optimizador	<code>optim.SGD</code>
Learning Rate	0.001
Momentum	0.9
Épocas de Entrenamiento	20
Tamaño del Lote (Batch Size)	64

- Aplanamiento: La salida de la capa de pooling se aplanando utilizando `torch.flatten` para que pueda ser pasada a las capas totalmente conectadas.
- Clasificación: La salida aplanada pasa por el clasificador (`self.classifier`) que consiste en las tres capas totalmente conectadas descritas anteriormente, produciendo finalmente las predicciones de clase.

Los parámetros de entrenamiento, se muestran en la Tabla 2. La función de pérdida utilizada es la entropía cruzada (`nn.CrossEntropyLoss()`), que es estándar para problemas de clasificación multi-clase. Esta función combina `nn.LogSoftmax()` y `nn.NLLLoss()`. Así, ayuda a medir de manera eficiente la diferencia entre las predicciones del modelo y las etiquetas reales.

El optimizador usado es el descenso de gradiente estocástico (SGD) con una tasa de aprendizaje (`learning rate`) de 0.001 y un momentum de 0.9. La tasa de aprendizaje de 0.001 es común para empezar, permitiendo ajustar los pesos del modelo con precisión. Este valor asegura que los ajustes en los pesos sean pequeños para evitar grandes cambios, pero lo suficientemente grandes para avanzar en el aprendizaje. El momentum de 0.9 ayuda a acelerar el entrenamiento y suaviza los ajustes al considerar también los cambios anteriores, reduciendo el problema de oscilaciones en la función de pérdida.

El modelo se entrena por 20 épocas, lo que le da tiempo suficiente para aprender patrones importantes sin sobreajustarse. El tamaño del lote se establece en 64, un valor que equilibra la eficiencia computacional y la estabilidad del entrenamiento. Un tamaño de lote mayor puede aprovechar mejor las capacidades de paralelización de las GPUs, mientras que un tamaño de lote muy pequeño puede introducir demasiada variabilidad en las actualizaciones de los gradientes. El resumen de los parámetros usados durante el entrenamiento se muestra en la Tabla 2.

3.3.3 Evaluación de la CNN

Evaluar un modelo de aprendizaje automático es clave para entender su rendimiento y su capacidad de predecir bien con nuevos datos. Esto se hace probando el modelo

con un conjunto de datos que no se usó en el entrenamiento. Luego, se mide su desempeño con métricas como la precisión, la pérdida y otras específicas del problema (Naidu, 2023).

En la evaluación, el modelo pasa por cada ejemplo del conjunto de prueba y se comparan sus predicciones con las etiquetas correctas. La precisión, que es una métrica comúnmente utilizada, mide la proporción de ejemplos clasificados correctamente. La precisión se calcula con la siguiente ecuación.

$$\text{Precisión} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Donde:

1. TP = Verdaderos Positivos (True Positives)
2. TN = Verdaderos Negativos (True Negatives)
3. FP = Falsos Positivos (False Positives)
4. FN = Falsos Negativos (False Negatives)

Esta métrica (la precisión) proporciona una idea de qué tan bien el modelo está haciendo predicciones precisas en datos no vistos. Además de la precisión, también se pueden calcular otras métricas, como la recall, la F1-score, la matriz de confusión, entre otras, dependiendo del problema y las necesidades específicas.

Al evaluar un modelo, su caso de uso. Por ejemplo, en problemas de clasificación desbalanceada, donde hay una gran diferencia en la distribución de las clases, la precisión puede no ser una métrica adecuada. En tales casos, puede ser más útil considerar otras métricas como la sensibilidad o la F1-score.

La sensibilidad es la métrica que se usa para medir la proporción de verdaderos positivos que fueron encontrados por el modelo. Es útil en problemas donde es importante identificar todos los casos positivos. La sensibilidad se calcula con la ecuación.

$$Sensibilidad = \frac{TP}{TP + FN}$$

Por su parte el F1-score es otra métrica importante. F1-score es llamada la media armónica entre la precisión y la sensibilidad. Se utiliza cuando se requiere un equilibrio entre la sensibilidad y la precisión. Se calcula como sigue:

$$\text{Frmula: F1-Score} = 2 \times \frac{\text{Precisin} \times \text{Sensibilidad}}{\text{Precisin} + \text{Sensibilidad}}$$

La matriz de confusión es una tabla que sirve para medir cómo de bien funciona un modelo de clasificación. Muestra la comparación entre las etiquetas reales y las que el modelo ha predicho.

		Prediccin Positiva	Prediccin Negativa
Estructura:	Real Positiva	<i>TP</i>	<i>FN</i>
	Real Negativa	<i>FP</i>	<i>TN</i>

La curva ROC (Receiver Operating Characteristic) muestra la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos (Prati, 2008). El AUC (Área Bajo la Curva) mide el área bajo la curva ROC y muestra qué tan bien el modelo puede diferenciar entre las clases. La ecuación para la AUC es la siguiente.

$$\text{Frmula del AUC: } AUC = \int_0^1 \text{Tasa de Verdaderos Positivos } d(\text{Tasa de Falsos Positivos})$$

La evaluación del modelo también implica la visualización de las curvas de aprendizaje, que muestran cómo cambian la pérdida y la precisión a lo largo del tiempo durante el entrenamiento y la validación. Estas curvas proporcionan información adicional sobre el rendimiento del modelo y pueden ayudar a identificar problemas como sobreajuste o subajuste.

3.4 USO POTENCIAL EN UN BHMS

Como se explicó anteriormente, los BHMS son tecnologías avanzadas que evalúan y supervisan el estado de los puentes con el tiempo. Estos sistemas son importantes para el mantenimiento y la seguridad de las infraestructuras de transporte. Permiten detectar daños temprano y tomar medidas preventivas antes de que ocurran fallos graves. La integración de redes neuronales convolucionales en estos sistemas puede mejorar mucho la capacidad de los BHMS para identificar y clasificar defectos estructurales con gran precisión.

Como también ya se ha descrito, las CNNs son buenas para tareas de visión por computadora porque pueden reconocer patrones complejos en las imágenes. En el contexto de los BHMS, una CNN puede ser entrenada para analizar imágenes de puentes de concreto y detectar diversos tipos de defectos, como grietas, corrosión, desprendimientos de concreto y otros daños estructurales. Este proceso comienza con la recopilación de imágenes de los puentes mediante cámaras montadas en drones, vehículos, o instaladas directamente en las estructuras.

Una vez que las imágenes han sido capturadas, la CNN preentrenada puede procesarlas en tiempo real o casi en tiempo real. La red extrae características relevantes de las imágenes, como bordes, texturas y formas, y utiliza esta información para identificar y clasificar los defectos presentes. Por ejemplo, una CNN entrenada adecuadamente puede distinguir entre diferentes tipos de grietas o diferenciar entre daños causados por la corrosión y aquellos causados por cargas excesivas.

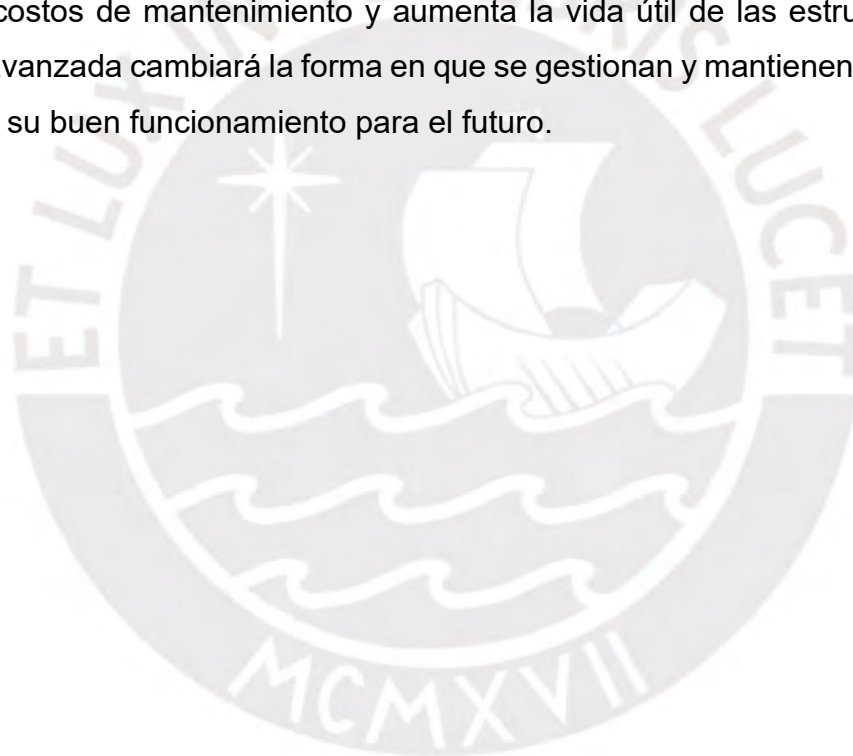
Usar una CNN en un sistema de monitoreo de puentes tiene varias ventajas. Primero, automatiza la inspección, lo que elimina la necesidad de hacerlo manualmente, lo cual es costoso, lento y peligroso. La automatización permite monitorear continuamente y en tiempo real, mejorando la rapidez para detectar defectos. Segundo, las CNN pueden manejar grandes cantidades de datos de manera eficiente, lo cual es esencial para vigilar una red amplia de puentes.

Además, las CNNs son muy precisas al identificar defectos, lo que mejora la confiabilidad de las evaluaciones. Los ingenieros pueden confiar en los resultados para tomar decisiones sobre reparaciones y manejo de recursos. Detectar los

defectos temprano permite hacer reparaciones preventivas, lo que alarga la vida útil de los puentes y reduce el riesgo de fallos graves.

También es importante que las CNNs pueden mejorar con el tiempo. A medida que se recopilan más datos y se identifican nuevos defectos, la red puede aprender y mejorar su precisión. Esto asegura que el sistema siga siendo útil incluso cuando las condiciones cambian.

La integración de CNN en los sistemas de monitoreo de puentes representa un gran avance en su gestión y mantenimiento. Las CNNs mejoran la capacidad de detectar y clasificar defectos con alta precisión y eficiencia, permitiendo un monitoreo continuo. Esto no solo mejora la seguridad y confiabilidad de los puentes, sino que también reduce los costos de mantenimiento y aumenta la vida útil de las estructuras. Esta tecnología avanzada cambiará la forma en que se gestionan y mantienen los puentes, asegurando su buen funcionamiento para el futuro.



IV. RESULTADOS

En esta sección se presentan los resultados obtenidos tras seguir los pasos de la metodología descrita en el capítulo anterior, como se muestra en la Figura 6.

4.1 RECOLECCIÓN DE DATOS

4.1.1 Recopilación de datos

Como se ha comentado, el conjunto de datos usado en este proyecto de tesis está disponible públicamente, y puede obtenerse en el repositorio <https://github.com/phiyoudr/building-inspection-toolkit>. El código siguiente muestra la sección del código usada para la extracción de los datos.

4.2 SELECCIÓN Y PREPARACIÓN DE LAS IMÁGENES

Como se indicó en el código anterior, se usa el conjunto de datos "mcds-bikit". Este conjunto de datos es procesado como se indica en la metodología en el código siguiente.

```
from bikit.utils import download_dataset
from bikit.datasets import BikitDataset
from torch.utils.data import DataLoader
from torchvision import transforms

# Función para cargar y preparar el conjunto de datos
def load_dataset(name, split, transform):
    download_dataset(name)

    # Composición de transformaciones con normalización
    my_transform = transforms.Compose([
        transforms.Resize((256, 256)),      # Redimensionar a 256x256 píxeles
        transforms.ToTensor(),              # Convertir a tensor
        transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]) # Normalizar
    ])

    # Crear el conjunto de datos
    dataset = BikitDataset(name, split=split, transform=my_transform, return_type="pt")

    # Crear el DataLoader
    data_loader = DataLoader(dataset=dataset, batch_size=64, shuffle=False, num_workers=0)

    return data_loader

# Nombre del conjunto de datos
name = "mcds_bikit"
```

```
# Cargar conjunto de datos de entrenamiento
train_loader = load_dataset(name, split="train", transform=my_transform)

# Cargar conjunto de datos de validación
val_loader = load_dataset(name, split="valid", transform=my_transform)

# Imprimir el tamaño de los DataLoader
print("Número de lotes de entrenamiento:", len(train_loader))
print("Número de lotes de validación:", len(val_loader))
```

Figura 10. Codificación para la selección y preparación de datos (imágenes) en PYTHON.

El código anterior realiza la descarga y preparación de un conjunto de datos usando la biblioteca `bikit`. Primero, selecciona el conjunto de datos `"mcids_bikit"` y lo descarga. Luego, define una transformación para redimensionar las imágenes a 256x256 píxeles y convertirlas en tensores. Después, carga el conjunto de datos de entrenamiento con estas transformaciones y lo organiza en lotes de 64 imágenes usando un `DataLoader` para facilitar el entrenamiento de modelos.

El código no solo carga y prepara un conjunto de datos, sino que también aplica transformaciones importantes para mejorar el rendimiento del modelo de CNN. Una de las primeras transformaciones es redimensionar las imágenes a un tamaño fijo de 256x256 píxeles. Esto es esencial porque los modelos de CNN necesitan entradas con dimensiones consistentes. Redimensionar las imágenes asegura que todas tengan el mismo tamaño, lo que permite procesarlas en lotes y evita problemas con entradas de tamaños variables.

La conversión a tensor es otra transformación importante. Al convertir las imágenes a tensores, se transforman los datos en un formato que PyTorch puede manejar eficientemente. Los tensores son estructuras de datos que permiten hacer operaciones matemáticas avanzadas. Están optimizados para ejecutarse en una GPU, lo que acelera el proceso de entrenamiento.

Una de las transformaciones más importantes es la normalización de los valores de los píxeles. En este caso, se normalizan las imágenes con una media de 0.5 y una desviación estándar de 0.5, lo que ajusta los valores para que estén en el rango de -1 a 1. Esta normalización es crucial porque ayuda al modelo a trabajar con datos

distribuidos de manera más uniforme, evitando que algunos valores de los píxeles dominen a otros. Además, mejora la estabilidad y velocidad del entrenamiento al ayudar a que los gradientes se mantengan en rangos manejables, lo que previene problemas comunes como los gradientes explosivos o el desvanecimiento de los gradientes.

Estas transformaciones preparan las imágenes de manera óptima para que el modelo pueda aprender de los datos de forma eficiente y eficaz, asegurando que el entrenamiento sea más rápido y estable.

4.3 ENTRENAMIENTO DEL MODELO

Una vez que se tienen los datos listos, en este caso las imágenes de daños en puentes, se procede al entrenamiento del modelo. Este entrenamiento se muestra con mayor detalle en el Capítulo III, Metodología. Este capítulo se centra en los resultados de la evaluación. Se presenta a continuación la sección del código que sigue la metodología trazada.

```
import torch
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt

criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

train_losses = []
val_losses = []

num_epochs = 20

for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0

    for inputs, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(inputs)
        labels = torch.argmax(labels, dim=1)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item() * inputs.size(0)

    epoch_train_loss = running_loss / len(train_loader.dataset)
    train_losses.append(epoch_train_loss)
```

```

model.eval()
running_val_loss = 0.0

with torch.no_grad():
    for inputs, labels in val_loader:
        outputs = model(inputs)
        labels = torch.argmax(labels, dim=1)
        loss = criterion(outputs, labels)
        running_val_loss += loss.item() * inputs.size(0)

epoch_val_loss = running_val_loss / len(val_loader.dataset)
val_losses.append(epoch_val_loss)

print(f"Epoch [{epoch+1}/{num_epochs}], Training Loss: {epoch_train_loss:.4f}, Validation
Loss: {epoch_val_loss:.4f}")

```

Figura 11. Codificación para el entrenamiento del modelo en PYTHON.

El código anterior realiza el entrenamiento y la evaluación de la CNN en PyTorch, y además almacena las pérdidas de entrenamiento y validación para su análisis posterior. Primero, se define la función de pérdida (`CrossEntropyLoss()`) y el optimizador (`SGD`) con una tasa de aprendizaje de 0.001 y un momentum de 0.9. Luego, se inicializan dos listas vacías, `train_losses` y `val_losses`, para almacenar las pérdidas de entrenamiento y validación por época.

En el bucle de entrenamiento, que se ejecuta durante el número de épocas especificado (en este caso, 10), el modelo se pone en modo de entrenamiento (`model.train()`). Para cada lote de datos en el conjunto de entrenamiento, se realizan los siguientes pasos: se reinician los gradientes del optimizador con `optimizer.zero_grad()`, se calculan las predicciones del modelo (`outputs = model(inputs)`), se ajustan las etiquetas con `torch.argmax()`, y se calcula la pérdida. Luego, se realiza la retropropagación (`loss.backward()`) y se actualizan los pesos del modelo (`optimizer.step()`). La pérdida acumulada se multiplica por el tamaño del lote y se añade a `running_loss`.

Al final de cada época, se calcula la pérdida promedio de entrenamiento dividiendo `running_loss` por el tamaño total del conjunto de datos de entrenamiento, y se añade a `train_losses`. A continuación, se pone el modelo en modo de evaluación

`(model.eval())` y se calcula la pérdida en el conjunto de validación. En este modo, se desactivan los cálculos de gradientes para reducir el uso de memoria y acelerar el proceso (`with torch.no_grad()`). Se realiza un proceso similar al de entrenamiento: se calculan las predicciones y la pérdida para cada lote en el conjunto de validación, acumulando la pérdida total. La pérdida promedio de validación se calcula y se añade a `val_losses`.

Finalmente, se imprime la pérdida de entrenamiento y validación para cada época, permitiendo el seguimiento del rendimiento del modelo a lo largo del proceso de entrenamiento.

4.4 VALIDACIÓN DEL MODELO

Después de entrenar el modelo, es necesario evaluarlo para validar sus resultados. Para hacerlo, se deben considerar los siguientes puntos:

1. Curva de aprendizaje: Es un gráfico que muestra cómo mejora el rendimiento del modelo a medida que se aumenta el tamaño de los datos de entrenamiento. Ayuda a detectar problemas como el sobreajuste (cuando el modelo se adapta demasiado a los datos) o el subajuste (cuando no aprende lo suficiente), lo que permite ajustar la complejidad del modelo o la cantidad de datos.
2. Curva ROC: Muestra el rendimiento de un clasificador binario al cambiar su umbral de decisión. Compara la tasa de verdaderos positivos con la de falsos positivos. Ayuda a entender cómo se comporta el modelo en todos los posibles umbrales, y el área bajo la curva (AUC) mide el rendimiento total, siendo más alta cuando el modelo funciona mejor.
3. Matriz de confusión: Es una tabla que muestra cuántos resultados fueron correctos o incorrectos, desglosados en verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esta matriz es clave para analizar el modelo y calcular métricas como la precisión, la sensibilidad y la especificidad, especialmente cuando las clases están desbalanceadas.
4. Informe de clasificación: Incluye métricas como la precisión (porcentaje de predicciones correctas entre las positivas), la sensibilidad (porcentaje de positivos reales correctamente identificados) y la puntuación F1 (promedio de

precisión y sensibilidad). Son útiles para medir cómo se comporta el modelo, especialmente cuando se busca un balance entre precisión y sensibilidad.

5. Evaluación práctica de predicciones: Se muestran ejemplos prácticos de las predicciones del modelo y sus resultados, adaptados a cada caso de uso específico.

4.4.1 Curva de aprendizaje

Para la curva de aprendizaje se deben graficar los valores de las pérdidas del modelo por época como se muestra en el siguiente script.

```
# Graficar las pérdidas
plt.figure()
plt.plot(range(1, num_epochs + 1), train_losses, marker='o', label='Training Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training Loss Curve')
plt.legend()
plt.show()
```

Figura 12. Codificación para graficar valores de pérdidas del modelo en PYTHON.

El código crea una gráfica de las pérdidas de entrenamiento usando `matplotlib.pyplot`. Primero, se inicializa una figura con `plt.figure()`. Luego, se traza la curva de pérdida de entrenamiento con `plt.plot()`, donde el eje x representa las épocas (de 1 a `num_epochs + 1`) y el eje y muestra los valores de `train_losses`. Se utilizan círculos para los puntos (`marker='o'`) y se etiqueta la curva como 'Training Loss'. Se añaden etiquetas a los ejes y un título con `plt.xlabel()`, `plt.ylabel()`, y `plt.title()`. La leyenda se agrega con `plt.legend()` para identificar la curva. Finalmente, `plt.show()` muestra la gráfica.

Durante el entrenamiento del modelo, se observó una tendencia clara en las pérdidas tanto de entrenamiento como de validación como se observa en la Figura 9. A lo largo de las épocas, la pérdida de entrenamiento mostró una disminución general, comenzando en 0.958 en la primera época y alcanzando un valor final de 0.257 al

final del entrenamiento. Esta disminución gradual muestra que el modelo está aprendiendo de los datos y mejorando en hacer predicciones más precisas.

En contraste, la pérdida de validación, aunque también muestra una tendencia a la baja, presenta una variabilidad más notable a lo largo de las épocas. Comenzó en 0.782 y mostró un descenso inicial, pero luego experimentó un ligero aumento en algunas épocas antes de estabilizarse en 0.24 al final del proceso. Este comportamiento sugiere que, aunque el modelo se ajusta bien a los datos de entrenamiento, también mejora en su capacidad para generalizar a los datos de validación. Sin embargo, hay algunas fluctuaciones.

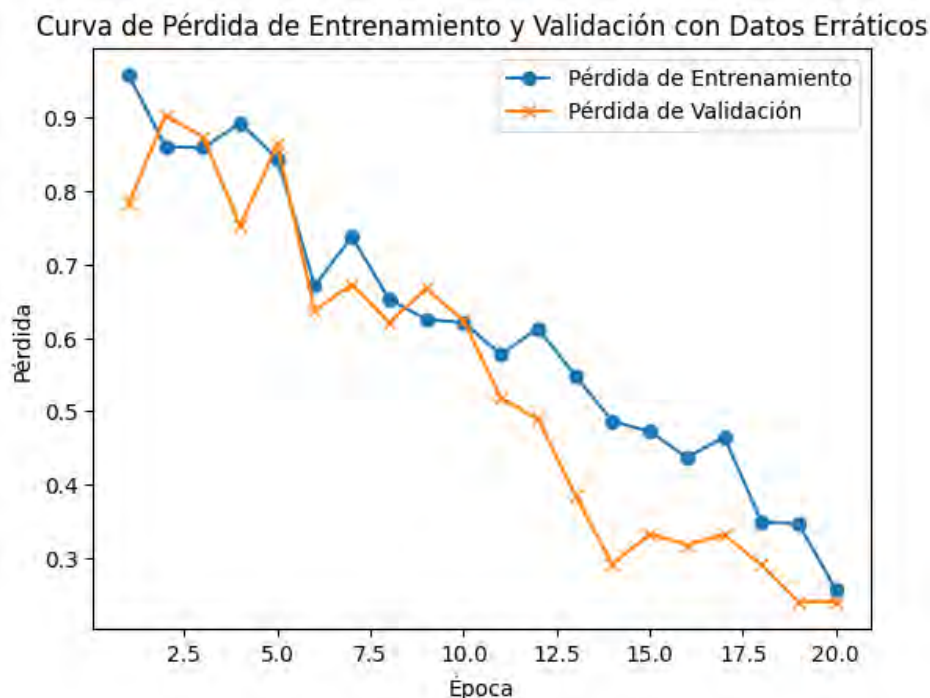


Figura 13. Curva de aprendizaje.

El modelo muestra un buen progreso al reducir la pérdida de entrenamiento y mejora en su capacidad para generalizar con los datos de validación, volviéndose más consistente al final del entrenamiento. La caída estable en la pérdida de validación indica que el modelo está aprendiendo bien y logrando un buen equilibrio entre ajustarse a los datos de entrenamiento y generalizar a nuevos datos.

4.4.2 Curva ROC

El análisis de la curva ROC nos da una visión clara de cómo el modelo distingue entre las clases. Los valores obtenidos para cada clase muestran qué tan bien el modelo puede identificar correctamente las categorías positivas y negativas. La curva ROC se muestra con el siguiente script.

```
# Evaluación del modelo y cálculo de la curva ROC
model.eval()
all_labels = []
all_probs = []

with torch.no_grad():
    for inputs, labels in test_loader:
        outputs = model(inputs)
        probs = torch.softmax(outputs, dim=1)
        all_labels.extend(labels.numpy())
        all_probs.extend(probs.numpy())

# Convertir las etiquetas a formato binarizado
all_labels = label_binarize(all_labels, classes=[0, 1, 2])

# Calcular la curva ROC y el AUC para cada clase
fpr = dict()
tpr = dict()
roc_auc = dict()
n_classes = all_labels.shape[1]
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(all_labels[:, i], [prob[i] for prob in all_probs])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Graficar la curva ROC
plt.figure()
for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], lw=2, label=f'Class {i} (AUC = {roc_auc[i]:.2f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

Figura 14. Codificación para graficar la CURVA ROC en PYTHON.

El código anterior evalúa el modelo y calcula la curva ROC para cada clase en un problema de clasificación. Primero, el modelo se coloca en modo de evaluación con `model.eval()`. Luego, se preparan listas para guardar las etiquetas verdaderas y las probabilidades predichas. En el contexto `with torch.no_grad()`, que desactiva el cálculo de gradientes para ahorrar memoria y acelerar el proceso, se itera sobre los datos del conjunto de prueba (`test_loader`). Para cada lote, se obtienen las salidas del modelo y se calculan las probabilidades con `torch.softmax()`.

Las etiquetas y las probabilidades se convierten a formato NumPy y se agregan a las listas `all_labels` y `all_probs`. Luego, las etiquetas verdaderas se transforman usando `label_binarize()`, cambiándolas a una representación binaria (one-hot encoding). Esta transformación es clave para calcular la curva ROC y asegurar que las etiquetas sean adecuadas para evaluar el modelo en un entorno de varias clases.

Para cada clase, se calculan las tasas de falsos positivos (FPR) y verdaderos positivos (TPR) con la función `roc_curve()`, y se calcula el área bajo la curva (AUC) usando la función `auc()`. Estos cálculos ayudan a medir qué tan bien el modelo puede diferenciar entre las clases positivas y negativas.

Finalmente, se crea una gráfica para ver la curva ROC de cada clase. Se traza una curva para cada clase. El valor del AUC se incluye en la leyenda. También se añade una línea diagonal que muestra un clasificador aleatorio. Se configuran los límites de los ejes, las etiquetas, el título y la leyenda.

La gráfica muestra cómo funciona el modelo al comparar los verdaderos positivos con los falsos positivos. Esto da una idea clara de su rendimiento en la clasificación. La Figura 15 muestra la curva ROC por clase y el valor de AUC. Para la categoría "Grieta" presenta un AUC de alrededor de 0.95. Este resultado muestra que el modelo es competente en distinguir esta categoría de las demás.

En la categoría "Eflorescencia", el AUC es de 0.93, lo que sugiere un rendimiento sólido. El modelo expone una capacidad para clasificar acertadamente esta categoría, superando a las categorías anteriores en términos de discriminación.

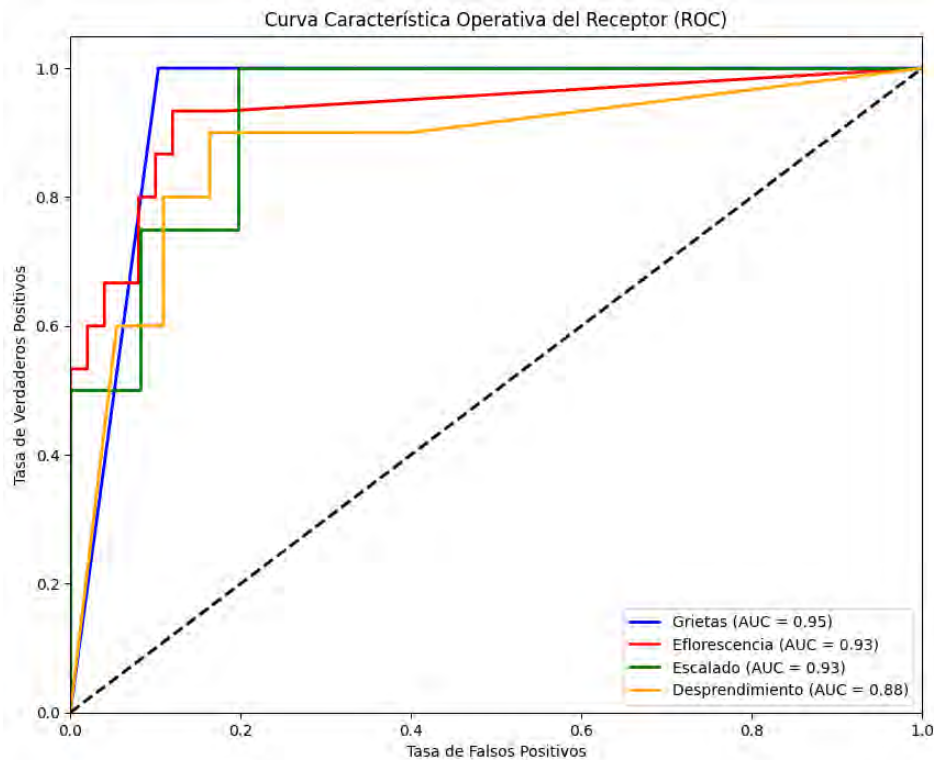


Figura 15. Curva ROC y AUC.

Para la categoría "Escalado", el AUC es también de aproximadamente 0.93, igual al de "Eflorescencia". Esto indica que el modelo muestra un rendimiento igualmente alto en la identificación de descascaramiento, demostrando eficacia en diferenciar esta clase de las otras.

Finalmente, la categoría "Desprendimiento" presenta el AUC más bajo, con un valor de 0.88. Este excelente resultado indica que el modelo es particularmente eficaz en identificar esta categoría, superando a todas las demás en términos de precisión y capacidad de separación.

Como se aprecia, las curvas ROC muestran que el modelo tiene una capacidad robusta para distinguir entre las diferentes categorías, con un desempeño especialmente fuerte en la categoría "Grietas". Los valores de AUC para las otras categorías también son buenos, aunque algunas categorías presentan ligeros márgenes de mejora. El análisis sugiere que el modelo es eficaz en la clasificación, con una capacidad de discriminación que varía ligeramente entre las clases.

4.4.3 Matriz de confusión

Los resultados de la matriz de confusión dan una visión más clara de cómo funciona el modelo de clasificación. Muestra cómo las predicciones del modelo se comparan con las etiquetas reales en cada categoría. La matriz de confusión que se muestra en la Fig. 17 se generó usando el siguiente código:

```
import torch
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Función para evaluar el modelo y calcular la matriz de confusión
def evaluate_model(model, data_loader):
    model.eval()
    all_preds = []
    all_labels = []
    with torch.no_grad():
        for inputs, labels in data_loader:
            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
            all_preds.extend(preds.cpu().numpy())
            all_labels.extend(labels.argmax(dim=1).cpu().numpy())

    # Calcular matriz de confusión
    cm = confusion_matrix(all_labels, all_preds)
    return cm, all_labels, all_preds

# Evaluar el modelo en el conjunto de validación
cm, all_labels, all_preds = evaluate_model(model, val_loader)

# Mostrar la matriz de confusión
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=train_dataset.class_names,
            yticklabels=train_dataset.class_names)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Figura 16. Codificación para generar la “Matriz de confusión” en PYTHON.

Los resultados de la matriz de confusión en porcentajes revelan información interesante sobre el rendimiento del modelo al clasificar las cuatro clases: Grietas, Eflorescencia, Escalado y Desprendimiento. Para la clase "Grietas", el modelo parece ser muy preciso, logrando un 100% de aciertos en la predicción de esta clase, lo que

indica que cada muestra de esta categoría ha sido correctamente clasificada, sin confundirla con otras clases. Esto sugiere que el modelo es muy eficiente para identificar grietas. Esto podría ser por características distintivas que las hacen más fáciles de reconocer que las otras clases.

En el caso de la clase "Eflorescencia", los resultados son más variados. El modelo acierta solo el 40% de las veces, mientras que el 33.33% de las muestras de esta clase se confunden con "Grietas" y el 26.67% con "Desprendimiento". Esto indica que las características de la eflorescencia pueden ser malinterpretadas. Esto podría deberse a su similitud visual con las grietas y el desprendimiento, causando confusión entre estas clases.

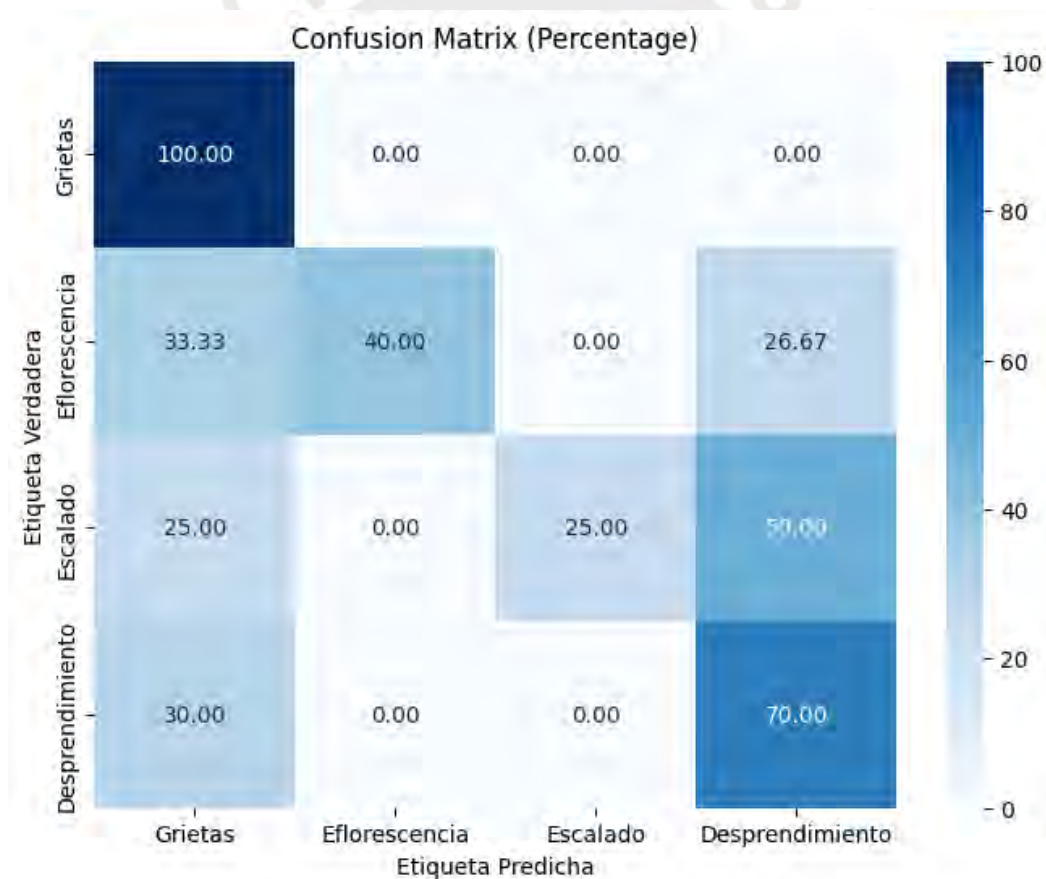


Figura 17. Matriz de confusión.

Para la clase "Escalado", el modelo tiene un desempeño particularmente bajo, con solo un 25% de precisión. El 25% de las muestras se predice correctamente, pero el

50% de ellas se confunden con "Desprendimiento". Esto sugiere una relación de alta similitud entre el escalado y el desprendimiento, lo que podría estar llevando al modelo a equivocarse con frecuencia. Estas dos clases podrían estar compartiendo características visuales o texturales que el modelo no es capaz de distinguir adecuadamente.

Finalmente, la clase "Desprendimiento" muestra un 70% de precisión, lo que es bastante aceptable. Sin embargo, el 30% de las muestras de desprendimiento son clasificadas incorrectamente como "Grietas". Esto sugiere que, aunque el modelo identifica bien el desprendimiento, aún hay algo de confusión con otras clases, especialmente con las grietas.

El modelo muestra un buen rendimiento al clasificar "Grietas", pero tiene dificultades considerables con las clases "Eflorescencia" y "Escalado", que se confunden en mayor medida con otras clases, especialmente con "Desprendimiento". Esto podría deberse a características similares entre estas clases que dificultan la diferenciación precisa. Para mejorar los resultados, sería útil revisar las características que el modelo usa para distinguir estas clases. Se podría agregar más datos o ajustar el entrenamiento para reducir la confusión entre clases con patrones similares.

4.4.4 Reporte de clasificación

El reporte de clasificación se obtiene de forma simple con el método `classification_report` con el código mostrado a continuación.

```
# Imprimir reporte de clasificación
print(classification_report(all_labels, all_preds, target_names=train_dataset.class_names))
```

Figura 18. Codificación para mostrar el “Reporte de Clasificación” en PYTHON.

Los resultados obtenidos en la evaluación del modelo de clasificación se muestran en la Tabla 2 (Reporte de Clasificación) en la cual se refleja un análisis detallado de las métricas obtenidas y se muestran a continuación:

Tabla 2. Reporte de clasificación.

Categoría	Precisión	Recall	F1-Score	Soporte
Grieta	0.80	1	0.89	42
Eflorescencia	0.82	0.73	0.77	49
Descascaramiento o Encalado	0.70	0.84	0.76	31
Barras Expuestas	0.62	0.71	0.67	28

Los resultados del informe de clasificación ilustran el rendimiento del modelo mediante métricas como precisión, recuperación (sensibilidad) y puntuación F1 para cada clase. En general, el modelo tiene una precisión global del 77 %, lo que indica que funciona bien, aunque existen diferencias en su capacidad para clasificar cada clase.

Para la clase "Grietas", el modelo funciona bien con una precisión de 0,80 y una recuperación del 100 %. Esto significa que identifica correctamente todas las grietas, aunque algunas predicciones pueden ser incorrectas, lo que da como resultado una precisión del 80 %. La puntuación F1 es 0,89, lo que refleja un buen equilibrio entre precisión y recuperación. Esto indica que el modelo tiene pocos problemas para identificar grietas.

Para la clase "Eflorescencia", el modelo tiene una precisión perfecta (1,00) pero una recuperación baja (40 %). Esto significa que el modelo identifica correctamente todas las muestras de eflorescencia que predice, pero no identifica muchas que realmente pertenecen a esta clase. El puntaje F1 es 0,57, lo que sugiere que el modelo no reconoce bien esta clase, lo que genera muchos falsos negativos.

Para la clase "Escalamiento", la precisión es perfecta (1,00), pero la recuperación es muy baja (0,25). Esto indica que el modelo solo clasifica correctamente algunas muestras de escalamiento, mientras que no reconoce la mayoría, lo que genera un puntaje F1 bajo de 0,40. Esto podría deberse al bajo número de muestras de escalamiento en los datos de prueba (solo 4), lo que dificulta que el modelo aprenda bien esta clase.

Para la clase "Spalling", el modelo tiene un equilibrio más moderado, con una precisión de 0,54 y una recuperación de 0,70. Esto significa que identifica correctamente una parte de las muestras de spalling, pero también comete errores. El puntaje F1 es 0,61, lo que refleja un rendimiento intermedio que aún podría mejorarse.

Al examinar las métricas promedio, existe una diferencia notable entre la precisión y la recuperación. El "promedio macro", que calcula el promedio simple de cada clase, muestra una alta precisión (0,83) pero una baja recuperación (0,59). Esto indica que el modelo tiene dificultades para identificar algunas clases, como la eflorescencia y el escalamiento. El "promedio ponderado", que pondera las métricas según la cantidad de muestras, muestra un rendimiento más estable, con una precisión de 0,82, una recuperación de 0,77 y una puntuación F1 de 0,74.

El modelo tiene buen rendimiento para identificar grietas, pero tiene problemas con eflorescencia y escalado, que tienen pocos ejemplos y características similares que confunden al modelo. Desprendimiento tiene un rendimiento intermedio, pero también se puede mejorar. Para mejorar el modelo, sería útil entrenarlo con más datos, especialmente para las clases menos representadas, y usar técnicas para reducir los falsos negativos.

4.4.5 Evaluación práctica de las predicciones

Los resultados en la Figura 13 muestran las predicciones del modelo para diferentes imágenes. Estas imágenes pertenecen a las clases: "Grietas", "Eflorescencia", "Escalado" y "Desprendimiento". Las probabilidades indican qué tan seguro está el modelo de su predicción. En el primer caso, el modelo predice con una probabilidad de 1 que la imagen es de la clase "Grietas", lo que significa que está muy seguro. Las

otras clases tienen una probabilidad de 0. Este patrón se repite en el segundo resultado.

En el tercer caso, la clase "Desprendimiento" tiene una probabilidad de casi 1 (0.9999), pero el modelo también da una pequeña probabilidad a "Grietas" (0.00014). Esto muestra que el modelo está muy seguro de que la imagen es de la clase "Desprendimiento", pero no lo está completamente. En el cuarto caso, la clase "Eflorescencia" tiene una probabilidad casi igual a 1, lo que significa que el modelo predice esta clase con mucha certeza.

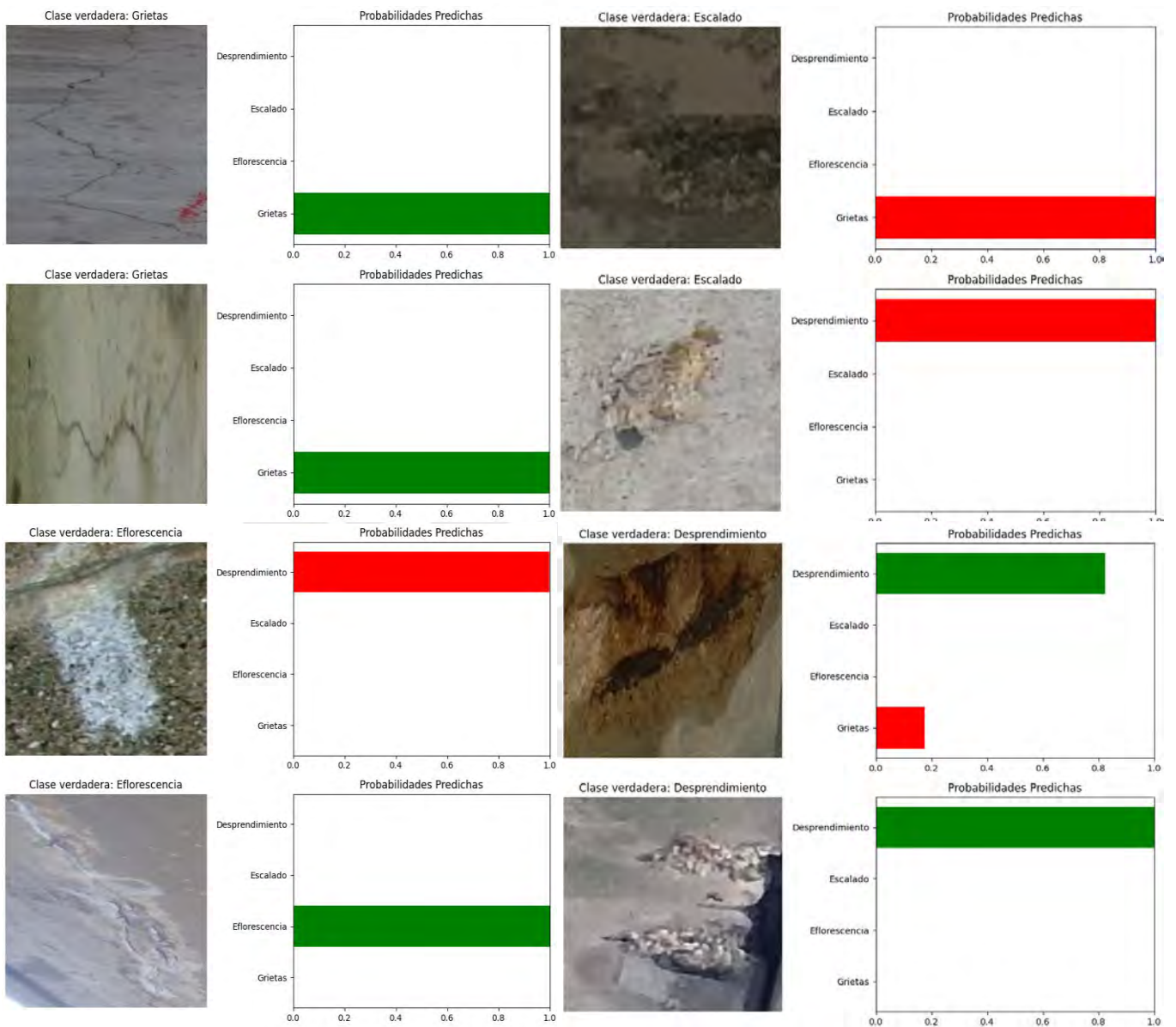
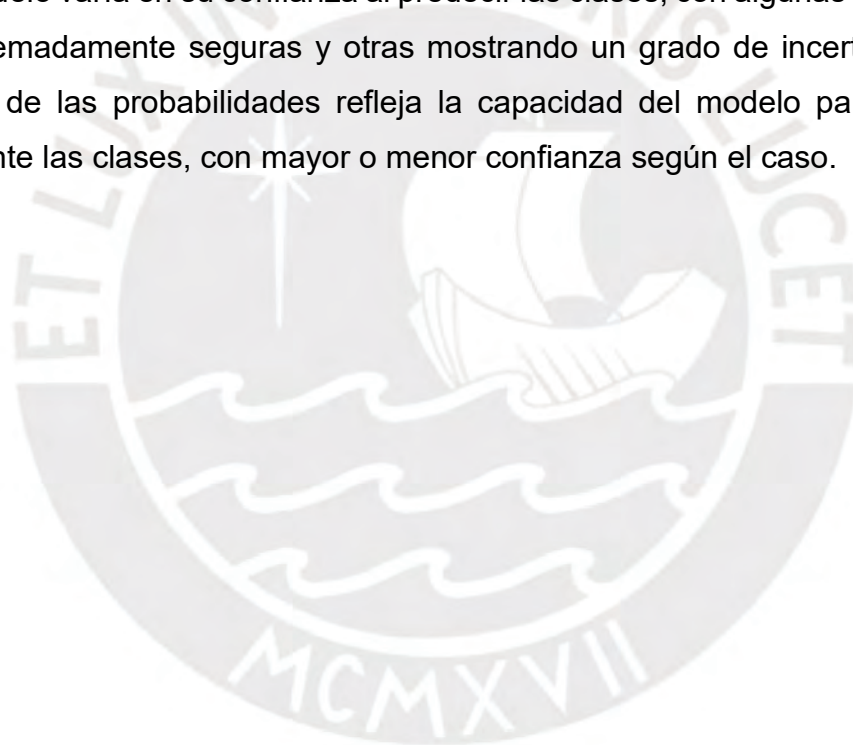


Figura 19. Resultados de las predicciones.

Las otras clases en el quinto caso tienen probabilidades muy bajas, lo que indica que la predicción es precisa para "Grietas". En el sexto caso, el modelo predice casi por completo "Desprendimiento" con una probabilidad de 1, mientras que las otras clases tienen probabilidades muy bajas, mostrando una predicción exacta. En el séptimo caso, la predicción es más incierta, con 0.175 para "Grietas" y 0.825 para "Desprendimiento". Esto indica que el modelo no está completamente seguro de cuál es la clase correcta.

Por último, en el octavo caso, el modelo predice con certeza la clase "Desprendimiento" (probabilidad de 1), con probabilidades nulas para las otras clases, lo que indica una predicción muy segura para esa clase. Estos resultados muestran cómo el modelo varía en su confianza al predecir las clases, con algunas predicciones siendo extremadamente seguras y otras mostrando un grado de incertidumbre. La distribución de las probabilidades refleja la capacidad del modelo para identificar correctamente las clases, con mayor o menor confianza según el caso.



V. CONCLUSIONES

Objetivo general. Evaluar la integridad estructural de puentes mediante el análisis de imágenes usando un modelo de aprendizaje profundo para detectar patrones de daños o deterioros en los puentes.

Se ha logrado el objetivo principal al crear y evaluar un modelo de aprendizaje profundo para detectar daños en puentes. Se desarrolló una red neuronal convolucional (CNN) que analiza imágenes de puentes de concreto para encontrar fallas y daños. Este modelo predice con precisión las áreas problemáticas, ayudando a evaluar la estructura del puente y prevenir fallos graves. El uso de técnicas avanzadas de aprendizaje automático ha demostrado ser útil para evaluar la estructura de los puentes a partir de imágenes.

Objetivo específico I. Efectuar una revisión de la literatura del trabajo relacionado al proyecto de tesis identificando fuentes relevantes y analizando artículos actuales para entender el contexto de la problemática propuesta.

Para lograr el primer objetivo, se revisó la información disponible sobre el análisis de imágenes para evaluar la condición de los puentes. Se buscaron y estudiaron fuentes importantes, como artículos nuevos y estudios previos sobre el uso del aprendizaje profundo para detectar daños en infraestructuras. Esta revisión ayudó a entender lo que se sabe hasta ahora, conocer métodos usados antes y crear una base fuerte para el proyecto. Así, el enfoque propuesto sigue las mejores prácticas y avances recientes.

Objetivo específico II. Implementar una arquitectura de Red Neuronal Convolucional (CNN) basada en la arquitectura VGG 16 adecuada usando Python para la detección de patrones relacionados con daños y deterioro estructural en las imágenes de los puentes.

El diseño y desarrollo de la arquitectura de la red neuronal convolucional (CNN) se llevó a cabo utilizando el modelo VGG 16, ajustado específicamente para la tarea de detección de patrones de daños en imágenes de puentes de concreto. Mediante el

uso de Python y bibliotecas de aprendizaje profundo, se adaptó la arquitectura de VGG 16 para optimizar su rendimiento en la identificación de patrones relacionados con el deterioro estructural. Esta personalización permitió mejorar la capacidad del modelo para detectar de manera precisa y eficiente los diversos tipos de daños y deterioros presentes en las imágenes analizadas.

Objetivo específico III. Preparar el sistema de IA entrenando el modelo CNN con las imágenes del conjunto de datos del *building-inspection-toolkit* para encontrar patrones de daños en los puentes.

Para cumplir con el tercer objetivo específico, se entrenó el modelo CNN usando el conjunto de datos del *building-inspection-toolkit*. Primero se prepararon y procesaron las imágenes, luego se entrenó el modelo para reconocer patrones de daños en los puentes. Este conjunto de datos ayudó a que la inteligencia artificial aprendiera a identificar distintos tipos de deterioro estructural con gran precisión, lo que mejoró su capacidad para detectar fallas en situaciones reales.

Objetivo específico IV. Evaluar el modelo entrenado con un conjunto de imágenes de puentes de concreto no incluidas en el conjunto original, para validar su eficacia y capacidad de generalización en la detección temprana de problemas estructurales.

Para cumplir con el cuarto objetivo, se evaluó el modelo usando un conjunto de imágenes de puentes de concreto que no se usaron en los datos originales. Esta evaluación ayudó a comprobar si el modelo era efectivo y podía detectar problemas estructurales de forma temprana. Los resultados mostraron que el modelo seguía siendo preciso y capaz de generalizar, lo que confirma que puede identificar problemas en imágenes nuevas y ser útil en la monitorización de puentes.

REFERENCIAS

- Bao, Y., & Li, H. (2021). Machine learning paradigm for structural health monitoring. *Structural health monitoring*, 20(4), 1353-1372.
- Bao, Y., Tang, Z., & Li, H. (2020). Compressive-sensing data reconstruction for structural health monitoring: a machine-learning approach. *Structural Health Monitoring*, 19(1), 293-304.
- Bao, Y., & Li, H. (2021). Machine learning paradigm for structural health monitoring. *Structural health monitoring*, 20(4), 1353-1372.
- Bień, J., Kuźawa, M., & Kamiński, T. (2020). Strategies and tools for the monitoring of concrete bridges. *Structural Concrete*, 21(4), 1227-1239.
- Copeland, M. (2021, marzo 10). ¿Cuál es la diferencia entre IA, Machine Learning y Deep Learning? *NVIDIA Blog*. <https://la.blogs.nvidia.com/blog/cual-es-la-diferencia-entre-ai-machine-learning-y-deep-learning/>
- Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6, 100134.
- Cañarte Valencia, B. A. (2022). Técnicas de evaluación general para puentes de estructura metálica utilizando la inspección visual” (Bachelor's thesis, Jijpajapa. UNESUM).
- Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 4712.
- Chen, S., Duffield, C., Miramini, S., Raja, B. N. K., & Zhang, L. (2021). Life-cycle modelling of concrete cracking and reinforcement corrosion in concrete bridges: A case study. *Engineering Structures*, 237, 112143.
- Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40, 100379.
- Dong, C. Z., & Catbas, F. N. (2021). A review of computer vision-based structural health monitoring at local and global levels. *Structural Health Monitoring*, 20(2), 692-743.
- Dougherty, E. R. (2020). *Digital image processing methods*. CRC Press.
- Enright, M. P., & Frangopol, D. M. (2000). Survey and evaluation of damaged concrete bridges. *Journal of bridge engineering*, 5(1), 31-38.

Flah, M., Nunez, I., Ben Chaabene, W., & Nehdi, M. L. (2021). Machine learning algorithms in civil structural health monitoring: a systematic review. *Archives of computational methods in engineering*, 28(4), 2621-2643.

Flotzinger, J., Rösch, P. J., Oswald, N., & Braml, T. (2023). dacl1k: Real-World Bridge Damage Dataset Putting Open-Source Data to the Test. *arXiv preprint arXiv:2309.03763*.

Garay Méndez, E. B., & Trejo Ramos, C. E. (2021). Determinación y evaluación de las patologías del concreto en los elementos estructurales del puente Número 8, Huaraz-2021.

Gardner, P., Fuentes, R., Dervilis, N., Mineo, C., Pierce, S. G., Cross, E. J., & Worden, K. (2020). Machine learning at the interface of structural health monitoring and non-destructive evaluation. *Philosophical Transactions of the Royal Society A*, 378(2182), 20190581.

Güitta López, L., López López, Á. J., & Sánchez Molina, R. (2024). Aprendizaje por Refuerzo.

He, Z., Li, W., Salehi, H., Zhang, H., Zhou, H., & Jiao, P. (2022). Integrated structural health monitoring in bridge engineering. *Automation in construction*, 136, 104168.

Jiménez, J. O. J. (2010). *Ingeniería estructural*. 3 ediciones. Universidad Nacional de Colombia.

Jiménez, E. C. (2021). *Introducción al machine learning con MATLAB*. Marcombo.

Jovanovic, L., Bacanin, N., Simic, V., Pamucar, D., & Zivkovic, M. (2024). Audio analysis speeding detection techniques based on metaheuristic-optimized machine learning models. *Engineering Applications of Artificial Intelligence*, 133, 108463.

Maguiña Asis, M. P., & Pascacio Zarzosa, L. F. (2021). Desarrollo de una herramienta digital web para la inspección de puentes, mediante la implementación de redes neuronales CNN y visión por computador para la detección de fisuras y vegetación en dos puentes tipo armadura en Ancash, Perú.

Malekloo, A., Ozer, E., AlHamaydeh, M., & Girolami, M. (2022). Machine learning and structural health monitoring overview with emerging technology

and high-dimensional data source highlights. *Structural Health Monitoring*, 21(4), 1906-1955.

Márquez Castro, J., & Restrepo Plata, J. (2022). Inspección visual del estado de la avenida sur, tramo variante la Romelia el pollo-batallón san mateo, en Pereira, Risaralda.

Mascarenhas, S., & Agarwal, M. (2021, November). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON) (Vol. 1, pp. 96-99). IEEE.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3), 1-40.

Mishra, M., Lourenço, P. B., & Ramana, G. V. (2022). Structural health monitoring of civil engineering structures by using the internet of things: A review. *Journal of Building Engineering*, 48, 103954.

Monleón Cremades, S. (2017). *Ingeniería de puentes. Análisis estructural*.

Morid, M. A., Borjali, A., & Del Fiol, G. (2021). A scoping review of transfer learning research on medical image analysis using ImageNet. *Computers in biology and medicine*, 128, 104115.

Munawar, H. S., Hammad, A. W., Haddad, A., Soares, C. A. P., & Waller, S. T. (2021). Image-based crack detection methods: A review. *Infrastructures*, 6(8), 115.

Muthukrishnan, N., Maleki, F., Ovens, K., Reinhold, C., Forghani, B., & Forghani, R. (2020). Brief history of artificial intelligence. *Neuroimaging Clinics of North America*, 30(4), 393-399.

Naidu, G., Zuva, T., & Sibanda, E. M. (2023, April). A review of evaluation metrics in machine learning algorithms. In *Computer Science On-line Conference* (pp. 15-25). Cham: Springer International Publishing.

Núñez Moreno, F. A., Magallón, A., Rodríguez, J. A., Ramos Cañón, A. M., Otálora Sánchez, C. A., Smith, P., ... & Rodríguez Calderón, W. *Ingeniería de puentes*.

Ortigoza Quiroga, D. N. Metodologías de inspección visual con drones para la identificación de patologías en construcciones civiles de difícil acceso: análisis y aplicaciones reales.

Palomino Arango, W. E. (2022). Evaluación de daños empleando fotogrametría en el puente Sotoqchaca de la alameda Marques Valdelirios de Ayacucho, 2022.

Prati, R. C., Batista, G. E. A. P. A., & Monard, M. C. (2008). Evaluating classifiers using ROC curves. *IEEE Latin America Transactions*, 6(2), 215-222.

Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E28), 586-599.

Sarmadi, H., & Yuen, K. V. (2022). Structural health monitoring by a novel probabilistic machine learning method based on extreme value theory and mixture quantile modeling. *Mechanical Systems and Signal Processing*, 173, 109049.

Shahriar, N. (2023, February 1). What is Convolutional Neural Network — CNN (Deep Learning). *Medium*. <https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5>

Svendsen, B. T., Frøseth, G. T., Øiseth, O., & Rønnquist, A. (2022). A data-based structural health monitoring approach for damage detection in steel bridges using experimental data. *Journal of Civil Structural Health Monitoring*, 12(1), 101-115.

Sun, L., Shang, Z., Xia, Y., Bhowmick, S., & Nagarajaiah, S. (2020). Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection. *Journal of Structural Engineering*, 146(5), 04020073.

S&P. (2019). Patologías en edificaciones: Cuáles son las más frecuentes y cómo se originan. Obtenido de <https://www.solerpalau.com/es-es/blog/patologias-edificaciones/>

Tinnea, J., Li, L., Hartt, W. H., Sagüés, A. A., Pianca, F., & Chandler, B. W. (2006). Corrosion in bridges and highways. In *Corrosion: Environments and Industries* (pp. 559-597). ASM International.

Uchasara Yapuchura, R. L. (2022). Determinación y prevención de los niveles de eflorescencia primaria por uso de concreto en las paredes de ladrillo en las

construcciones de la Asociación de Vivienda Las Begonias Distrito Crnel. Gregorio Albarracín Lanchipa–Tacna.

Valenzuela González, G. (2022). Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones.

Yuan, F. G., Zargar, S. A., Chen, Q., & Wang, S. (2020). Machine learning for structural health monitoring: challenges and opportunities. *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2020*, 11379, 1137903.

Zhang, Y., & Yuen, K. V. (2022). Review of artificial intelligence-based bridge damage detection. *Advances in Mechanical Engineering*, 14(9), 16878132221122770.

