UNIVERSIDAD NACIONAL DE MOQUEGUA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

"DIAGNÓSTICO DEL GRADO DE OSTEOFITO EN RADIOGRAFÍAS MEDIANTE UN MODELO HÍBRIDO DE MACHINE LEARNING"

TESIS

PRESENTADO POR:

PETER GIANFRANCO NEGRON YUCRA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMÁTICA

MOQUEGUA – PERÚ
2025

UNAM
UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Folio **Nº 022**

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución de Consejo de Facultad N° 171-2025-CF-FAIN-UNAM

Fecha: 06-10-2025

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

"Diagnóstico del grado de osteolito en radiografías mediante un modelo híbrido de Machine Learning"

Bachiller: Peter Gianfranco Negrón Yucra

Asesor Principal: Hugo Euler Tito Chura

Co-Asesor: .....

IV. Se dictamina:

Como: Aprobado por Unanimidad la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: Dieciseis (16) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA .

Siendo las 12:25 horas, del 09 de octubre del año 2025 se procede con la firma en conformidad de lo dictaminado por el Jurado.

Anibal Flores García
Presidente

Victor Yana Mamani
Primer Miembro

Salomón Rey Ramos Rivera
Segundo Miembro

Peter Negron Yucra
Bachiller

DEDICATORIA

Esta tesis la dedico a Dios por ser el faro que iluminó cada paso de este camino académico y por darme la fortaleza en los momentos de duda y temor.

A mi familia, quienes conté con su esfuerzo y apoyo incondicional durante cada etapa de mi vida.

A Alessandra, mi pareja, por su paciencia infinita y soporte inquebrantable durante largos días de trabajo.

Y a Princesa, que me acompaño durante momentos difíciles, aunque ya no esté físicamente, su huella permanece en cada página de esta tesis.

AGRADECIMIENTO

Agradezco a Dios por ser mi fortaleza durante este proceso, permitiéndome superar cada desafío con sabiduría y perseverancia.

A mi familia, cuyo apoyo constante me permitió alcanzar esta meta. Gracias por creer en mí incluso en los momentos más difíciles.

A Alessandra, por su paciencia, comprensión y por ser mi fuente de motivación y alegría en cada etapa de este camino.

A Princesa, cuya compañía y amor incondicional dejaron una huella en mi corazón y en este logro.

De manera especial, quiero expresar mi más sincero agradecimiento al Mgr. Euler Tito, por sus consejos que fueron clave para la culminación de esta investigación.

Finalmente, a todas las personas que de alguna manera formaron parte de este recorrido, mi más genuino agradecimiento.

RESUMEN

La investigación tiene el objetivo de diagnosticar y clasificar el grado de osteofitos en radiografías utilizando un modelo híbrido de Machine Learning. Esta arquitectura integra una red neuronal convolucional pre-entrenada para la extracción de características de imágenes y clasificadores de Gradient Boosting para la predicción del grado de osteofitos. Se implementó una estrategia de clasificación jerárquica que divide el problema en un modelo binario para diferenciar casos "No Severo" de "Severo", y un modelo multiclase para categorizar los grados severos específicos. Para mejorar la robustez y precisión, se aplicó un ensamble por votación suave entre los clasificadores; se ejecutó múltiples veces para hallar su media y desviación estándar.

Los resultados demuestran la superioridad de este enfoque. Los clasificadores alcanzaron una precisión global notable. Esta arquitectura jerárquica superó consistentemente a los modelos monolíticos que clasificaban directamente en todas las clases, los cuales presentaban dificultades con las clases minoritarias. El modelo híbrido propuesto ofrece una solución eficiente y precisa para el diagnóstico y gradación de osteofitos, con viabilidad para entornos con recursos computacionales limitados.

Palabras clave: Osteofitos, Machine Learning, Modelo Híbrido, Gradient Boosting.

ABSTRACT

The research aims to diagnose and classify the degree of osteophytes in X-rays using a hybrid machine learning model. This architecture integrates a pre-trained convolutional neural network for image feature extraction and Gradient Boosting classifiers for predicting the degree of osteophytes. A hierarchical classification strategy was implemented that divides the problem into a binary model to differentiate “Non-Severe” from “Severe” cases, and a multi-class model to categorize specific severe degrees. To improve robustness and accuracy, a soft voting ensemble was applied between the classifiers; it was run multiple times to find its mean and standard deviation.

The results demonstrate the superiority of this approach. The classifiers achieved remarkable overall accuracy. This hierarchical architecture consistently outperformed monolithic models that classified directly into all classes, which had difficulties with minority classes. The proposed hybrid model offers an efficient and accurate solution for the diagnosis and grading of osteophytes, with viability for environments with limited computational resources.

Keywords: Osteophytes, Machine Learning, Hybrid Model, Gradient Boosting.

ÍNDICE

AGRADECIMIENTO III
RESUMEN IV
ABSTRACT V
ÍNDICE VI
ÍNDICE DE TABLAS IX
ÍNDICE DE FIGURAS X
INTRODUCCIÓN XIII
1. PLANTEAMIENTO DEL PROBLEMA XIV
1.1. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA XIV
1.1.1. DESCRIPCIÓN DEL PROBLEMA XIV
1.1.2. FORMULACIÓN DEL PROBLEMA XV
1.1.2.1. INTERROGANTE PRINCIPAL XV
1.1.2.2. INTERROGANTES ESPECÍFICOS XVI
1.2. OBJETIVOS XVI
1.2.1. OBJETIVO GENERAL XVI
1.2.2. OBJETIVOS ESPECÍFICOS XVI
1.3. JUSTIFICACIÓN E IMPORTANCIA XVII
1.4. HIPÓTESIS XVII
1.4.1. HIPÓTESIS GENERAL XVII
1.4.2. HIPÓTESIS ESPECÍFICAS XVIII
2. MARCO TEÓRICO XVIII
2.1. ANTECEDENTES XVIII
2.2. BASES TEÓRICAS XXI
2.2.1. ANOMALÍAS OSEAS XXI
2.2.1.1. OSTEOFITOS XXI
2.2.1.2. OSTEOARTRITIS XXII
2.2.2. MACHINE LEARNING XXIV
2.2.2.1. DEFINICIÓN XXIV
2.2.3. REDES NEURONALES ARTIFICIALES XXV
2.2.3.1. DEFINICIÓN XXV
2.2.3.2. DEEP LEARNING XXVI
2.2.3.3. REDES NEURONALES CONVOLUCIONALES XXVII
2.2.4. MÉTODOS DE ENSAMBLAJE XXVIII
2.2.4.1. DEFINICIÓN XXVIII
2.2.4.2. GRADIENT BOOSTING XXIX

2.2.5. MODELOS HÍBRIDOS DE DEEP LEARNING Y MACHINE LEARNING XXX
2.3. DEFINICIÓN DE TÉRMINOS XXXIII
3. METODOLOGÍA XXXVII
3.1. TIPO Y NIVEL DE INVESTIGACIÓN XXXVII
3.2. ÁMBITO TEMPORAL Y ESPACIAL XXXVIII
3.3. POBLACIÓN Y MUESTRA XXXVIII
3.3.1. POBLACIÓN XXXVIII
3.3.2. MUESTRA XXXVIII
3.4. INSTRUMENTOS XXXIX
3.4.1. INSTRUMENTOS DE CÓMPUTO XXXIX
3.4.1.1. HARDWARE XXXIX
3.4.1.2. SOFTWARE XXXIX
3.4.2. INSTRUMENTOS CONCEPTUALES Y TÉCNICAS XLI
3.4.2.1. CLASIFICACIÓN JERÁRQUICA XLI
3.4.2.2. TRANSFERN LEARNING XLI
3.4.2.3. MODELADO HÍBRIDO XLII
3.4.2.4. MANEJO DE DESBALANCE XLII
3.4.2.5. TÉCNICA DE ENSAMBLE VOTING XLII
3.4.2.6. ECUACIONES DE MÉTRICAS DE EVALUACIÓN XLIII
3.5. PROCEDIMIENTOS XLIV
3.6. ANÁLISIS DE DATOS XLV
3.6.1. RECOLECCIÓN Y ESTRUCTURACIÓN DEL DATASET XLV
3.6.1.1. FUENTE Y CONTENIDO XLV
3.6.1.2. PARTICIÓN DE DATOS XLVI
3.6.2. DISEÑO DE LA CLASIFICACIÓN JERÁRQUICA XLVII
3.6.2.1. CLASIFICACIÓN JERARQUICA XLVII
3.6.3. PRE-PROCESAMIENTO DE DATOS XLVIII
3.6.3.1. PRE-PROCESAMIENTO DEL DATASET XLVIII
3.6.3.2. CÁLCULO DE PESOS XLIX
3.6.3.3. AUMENTACIÓN DE DATOS L
3.6.4. DISEÑO Y OPTIMIZACIÓN DEL EXTRACTOR DE CARACTERÍSTICAS LI
3.6.4.1. ANÁLISIS COMPARATIVO Y SELECCIÓN DEL MODELO BASE LI
3.6.4.2. IMPLEMENTACIÓN DE ESTRATEGIAS DE REGULARIZACIÓN LIII
3.6.4.3. OPTIMIZACIÓN DE HIPERPARÁMETROS LIV

3.6.4.4. CALLBACKS LV
3.6.4.5. FINE-TUNING LVI
3.6.5. ENTRENAMIENTO Y EVALUACIÓN DE CLASIFICADORES LVII
3.6.5.1. EXTRACCIÓN DE CARACTERÍSTICAS LVII
3.6.5.2. MANEJO DE DESBALANCE LVIII
3.6.5.3. EVALUACIÓN DE CLASIFICADORES LVIII
3.6.6. DESARROLLO DEL ENSAMBLE DE MODELOS LXI
3.6.7. ANÁLISIS DE ROBUSTEZ Y ESTABILIDAD DEL MODELO PROPUESTO LXII
3.6.7.1. PROCEDIMIENTO DE ROBUSTEZ LXII
3.6.7.2. ANÁLISIS ESTADÍSTICO DEL MODELO LXIII
3.6.8. DISEÑO DEL MODELO DE PREDICCIÓN FINAL LXIV
4. RESULTADOS Y DISCUSIÓN LXVI
4.1. RENDIMIENTO DEL MODELO DE REFERENCIA LXVI
4.2. RESULTADOS DE LA CLASIFICACIÓN JERÁRQUICA LXVIII
4.2.1. RENDIMIENTO INICIAL DEL MODELO 1 (BINARIO) LXVIII
4.2.2. RENDIMIENTO INICIAL DEL MODELO 2 (SEVERO) LXIX
4.3. RESULTADOS DE LOS MODELOS HÍBRIDOS LXX
4.3.1. COMPARACIÓN DE ARQUITECTURAS CNN LXX
4.3.2. COMPARACIÓN DE CLASIFICADORES DE GRADIENT BOOSTING LXXI
4.4. RENDIMIENTO DE MÉTRICAS Y ETIQUETAS LXXIII
4.5. ANÁLISIS DE ERRORES Y CASOS MAL CLASIFICADOS LXXVII
4.6. EVALUACIÓN FINAL DE LOS MODELOS LXXX
4.7. RESULTADOS DEL ANÁLISIS DE ROBUSTEZ LXXXII
4.8. COMPARACIÓN CON EL ESTADO DEL ARTE LXXXV
4.9. LIMITACIONES DEL ESTUDIO LXXXVII
5. CONCLUSIONES LXXXIX
5.1. CONCLUSIÓN GENERAL LXXXIX
5.2. CONCLUSIONES ESPECÍFICAS LXXXIX
6. RECOMENDACIONES XCII
7. REFERENCIAS XCIV
8. ANEXOS C
8.1. MATRIZ DE CONSISTENCIA C

ÍNDICE DE TABLAS

Tabla 1 Cantidad de imágenes del Dataset XLV

Tabla 2 Distribución de imágenes del Modelo Binario XLVII

Tabla 3 Distribución de imágenes del Modelo Severo XLVIII

Tabla 4 Comparación de Modelos Base para entrenamiento..... LII

Tabla 5 Resultados de los modelos CNN iniciales LXVI

Tabla 6 Resultados máximos de todos los modelos CNN aplicados LXXI

Tabla 7 Resultados máximos de los clasificadores de Gradient Boosting.....LXXII

Tabla 8 Resultados máximos de reportes de clasificación del modelo binario LXXIII

Tabla 9 Resultados máximos de las matrices de confusión del modelo binario LXXIV

Tabla 10 Resultados máximos de reportes de clasificación del modelo severo LXXV

Tabla 11 Resultados máximos de las matrices de confusión del modelo severo LXXVI

Tabla 12 Resultados del análisis de robustez de los modelos propuestos...LXXXII

Tabla 13 Comparación de modelos del estado del arte..... LXXXVI

ÍNDICE DE FIGURAS

Figura 1 Comparación de rodilla sana y rodilla con osteofitos XXII

Figura 2 Comparación entre articulación sana y con osteoartritis XXIII

Figura 3 Lectura de radiografía XXIV

Figura 4 Ejemplo sobre Machine Learning XXV

Figura 5 Esquema de red neuronal artificial XXV

Figura 6 Ejemplo de Deep Learning XXVI

Figura 7 Diseño de una red neuronal convolucional XXVII

Figura 8 Ejemplo de una operación de convolución XXVIII

Figura 9 Ejemplo de método de ensamblaje: Voting XXIX

Figura 10 Ejemplo de Gradient Boosting XXX

Figura 11 Enfoque de modelo híbrido de Machine Learning XXXI

Figura 12 Diagrama de flujo del pipeline de un modelo híbrido XXXII

Figura 13 Aplicación de Transfer Learning a ImageNet XXXIII

Figura 14 Exploración del dataset XLVI

Figura 15 Código de carga y optimización del dataset XLIX

Figura 16 Código de cálculo de pesos de clases L

Figura 17 Código de data augmentation LI

Figura 18 Código del modelamiento LIII

Figura 19 Código de regularización LIII

Figura 20 Código de búsqueda de hiperparámetros ideales LIV

Figura 21 Código de callbacks LV

Figura 22 Código de reentrenamiento con Fine-Tuning LVI

Figura 23 Código del extractor de características LVII

Figura 24 Código de sobremuestreo LVIII

Figura 25 Código del clasificador XGBoost LIX

Figura 26 Código del clasificador LightGBM LIX

Figura 27 Código del clasificador CatBoost LX

Figura 28 Código de reporte de clasificación y matriz de confusión..... LX

Figura 29 Código de evaluación de exactitud LXI

Figura 30 Código del ensamble de modelos binario y severo LXII

Figura 31 Lista de robustez en modelos entrenados LXIII

Figura 32 Código de cálculo de media y desviación estándar LXIV

Figura 33 Código de carga de componentes LXV

Figura 34 Diagrama de flujo del modelo de predicción final LXV

Figura 35 Monitoreo del entrenamiento con modelo ResNet50 LXVII

Figura 36 Monitoreo del entrenamiento con modelo EfficientNetB1 LXVII

Figura 37 Matriz de confusión del entrenamiento con EfficientNetB1 LXVIII

Figura 38 Reporte de clasificación y matriz de confusión del entrenamiento inicial con modelo binario..... LXIX

Figura 39 Reporte de clasificación y matriz de confusión del entrenamiento inicial con modelo severo..... LXX

Figura 40 Reporte de clasificación y matriz de confusión del entrenamiento ideal con modelo binario..... LXXIV

Figura 41 Reporte de clasificación y matriz de confusión del entrenamiento ideal con modelo severo..... LXXVI

Figura 42 Diagnóstico de Grado 2 en una radiografía de Grado 3 LXXVIII

Figura 43 Diagnóstico de Grado 4 en una radiografía de Grado 3 LXXVIII

Figura 44 Muestras de radiografías de los 5 Grados de Osteofitos..... LXXIX

Figura 45 Muestras de radiografías de Grados 2, 3 y 4 LXXIX

Figura 46 Resultado del diagnóstico final del modelo binario LXXX

Figura 47 Resultado del diagnóstico final del modelo binario y severo LXXXI

Figura 48 Comparación de robustez del modelo binario LXXXIII

Figura 49 Comparación de ejecuciones del modelo binario LXXXIII

Figura 50 Comparación de robustez del modelo severo LXXXIV

Figura 51 Comparación de ejecuciones del modelo severo LXXXIV

INTRODUCCIÓN

Los osteofitos son crecimientos óseos anormales en las articulaciones, esto representa un problema en el sector de la salud. Esta enfermedad involucra distintas partes del sistema óseo, como las rodillas, que causan dolores, pérdida de movimiento, comprimir las raíces nerviosas, produciendo debilidad o entumecimiento. Esta enfermedad se puede diagnosticar mediante radiografías, pero los síntomas no suelen presentarse hasta que se encuentre en grados severos.

La importancia de abordar este problema en el Perú, donde hay presencia de artritis, una anomalía que hace a las personas más propensas a desarrollar osteofitos, está en aumento. Esto marca la urgencia de métodos efectivos para el diagnóstico temprano y una clasificación precisa de la severidad de los osteofitos.

Debido a ello, la presente investigación propone un modelo híbrido de Machine Learning para el diagnóstico del grado de osteofitos en radiografías, según la escala de Kellgren-Lawrence (KL). El objetivo principal es implementar un modelo híbrido que no solo diagnostique, sino que también clasifique los diferentes grados de osteofitos con alta precisión, buscando así una manera óptima y ofrecer una herramienta aplicable en el sector salud.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

1.1.1. DESCRIPCIÓN DEL PROBLEMA

Los osteofitos son crecimientos anormales que involucran distintas partes del sistema óseo, mayormente afectan las articulaciones, provocando dolores, pérdida de movimiento, entre otras afecciones. Su diagnóstico comúnmente se realiza a través de radiografías, ya que sus signos o síntomas no son advertidos hasta sentir severos malestares.

Para el caso de la rodilla, el crecimiento de osteofitos reduce el espacio articular y puede conducir a la compresión de estructuras adyacentes. En casos avanzados, pueden contribuir a la rigidez, al bloqueo mecánico de la articulación y a un deterioro significativo de la función motriz.

Esta anomalía se presenta en la mayoría de las personas que sufren de artritis, por ende, una persona que padece esta enfermedad es más propensa a tener osteofitos, sin su diagnóstico temprano, puede generar problemas en la salud que afecten en su actividad física o acciones que requieran fuerza. (Mayo Clinic, 2022).

Según El equipo de la torre – Neurocirugía: “Los osteofitos no suelen causar síntomas y no se suelen detectar hasta que se realiza alguna prueba por imágenes. Sin embargo, hay algunos casos en que provocan dolor y limitan el movimiento en las articulaciones.”

En el Perú, se indica que, por cada hombre, seis mujeres padecen este problema de salud. Además, se estima desde el 2019, que cada año aumentan 100 los casos de artritis. (MINSA, 2019).

En 2017, se registraron alrededor de 300 casos de artritis en infantes en el Instituto Nacional de Salud del Niño. (AméricaEconomía, 2020).

Frente a todo ello, se ha decidido realizar una solución ante la problemática de la salud en el Perú usando técnicas de Inteligencia Artificial para el diagnóstico temprano de osteofitos según el grado Kellgren-Lawrence (KL) y así poder reducir las consecuencias de la enfermedad que no hayan sido vistas a tiempo. Esta propuesta está dirigida al sector de la salud, donde puede ser aplicable en diferentes países, debido a que los osteofitos se dan a nivel global. Esta investigación se llevará a cabo con conjuntos de datos libres e imágenes otorgadas por una institución dedicada al estudio y análisis de osteoartritis, se realizarán distintas muestras para evaluar la exactitud del modelo propuesto y mostrar la exactitud óptima que puede diagnosticar los grados de osteofitos.

1.1.2. FORMULACIÓN DEL PROBLEMA

1.1.2.1. INTERROGANTE PRINCIPAL

¿Puede un modelo híbrido de Machine Learning diagnosticar diferentes grados de osteofitos en radiografías?

1.1.2.2. INTERROGANTES ESPECÍFICOS

- ¿Con una arquitectura óptima de un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías?
- ¿Es el modelo propuesto significativamente mejor que los modelos del estado del arte para diagnosticar diferentes grados de osteofitos en radiografías?

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Implementar un modelo híbrido de Machine Learning para diagnosticar diferentes grados de osteofitos en radiografías.

1.2.2. OBJETIVOS ESPECÍFICOS

- Diseñar una arquitectura óptima de un modelo híbrido de Machine Learning para diagnosticar diferentes grados de osteofitos en radiografías.
- Evaluar y comparar los modelos del estado del arte con el modelo propuesto para diagnosticar diferentes grados de osteofitos en radiografías.

1.3. JUSTIFICACIÓN E IMPORTANCIA

En este proyecto de investigación se justifica económicamente, dado a que el investigador presenta los recursos monetarios requeridos para la elaboración de las diversas acciones del estudio.

Para el sector de salud, podrá ser favorecida a poder implementar este sistema de diagnóstico de enfermedades degenerativas como son los osteofitos, ya no habrá retención en la producción de radiografías para su revisión y diagnóstico, así mismo se evitarán los errores en los diagnósticos causados por confusiones en las radiografías.

En el aspecto social, beneficiará a la población que se vea afectada por la enfermedad y a aquellas personas que deseen conocer su estado de salud, ya que los resultados se publicarán a través de un informe de tesis en diferentes repositorios y revistas indexadas.

En el aspecto tecnológico, se dispondrá de la información necesaria como los datos de radiografías, técnicas de modelado, uso de arquitecturas, para llevar a cabo la investigación, demostrando de esta manera, los resultados deseados.

1.4. HIPÓTESIS

1.4.1. HIPÓTESIS GENERAL

Con un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías.

1.4.2. HIPÓTESIS ESPECÍFICAS

- Con una arquitectura óptima de un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías.
- Con respecto a los modelos del estado del arte, el modelo propuesto tiene rendimiento significativo para diagnosticar diferentes grados de osteofitos en radiografías.

2. MARCO TEÓRICO

2.1. ANTECEDENTES

Kokkotis et al. (2022) utilizaron modelos de machine learning: Random Forest, Multilayer Perceptron, Logistic Regression, Support Vector Machines y K-Nearest Neighbors para el diagnóstico de osteoartritis de rodilla. Los resultados de precisión obtenidos fueron, respectivamente: 73.55%, 73.20%, 73.27%, 73.36% y 71.55%.

Tiulpin et al. (2018) aplicaron redes neuronales convolucionales para el diagnóstico automático de osteoartritis obtuvieron un promedio de precisión del 66.71%.

Antony et al. (2016) utilizaron redes neuronales convolucionales para cuantificar el grado de osteoartritis, teniendo como resultado un promedio de 59.6%.

Wang et al. (2021) aplicaron un modelo de clasificación basado en una red neuronal convolucional para el diagnóstico automático de osteoartritis en la rodilla, logrando un promedio de precisión del 69.18%.

Tycowicz et al. (2020) utilizaron un modelo de clasificación de osteoartritis basada en redes neuronales de gráficos (GCN) obteniendo un promedio de precisión del 64%.

Durán et al. (2022) aplicaron un modelo de redes neuronales convolucionales para clasificar la severidad de osteoartritis. El modelo obtuvo una precisión del 72%.

Yang et al. (2024) plantearon un modelo de red neuronal convolucional y una red de grafos para la predicción automática de la severidad de la osteoartritis, logrando una precisión de clasificación del 73.8%.

Bengani et al. (2024) aplicaron un modelo híbrido de Deep Learning para predecir el precio de las acciones de Google, el cual integra Random Forest para la extracción de características y una red LSTM (Long Short-Term Memory) para la predicción. Este modelo híbrido logró un RMSE (Error Cuadrático Medio de la Raíz) de 7.13, demostrando ser más consistente y fiable en sus predicciones que los modelos tradicionales de Machine Learning (Random Forest con RMSE de 9.56) o Deep Learning (LSTM) por sí solos.

Mavaie et al. (2023) formularon un enfoque de modelo híbrido, donde entrena una red neuronal convolucional para la extracción de características y luego utiliza estas características extraídas como entrada para un clasificador de Machine Learning, XGBoost, para realizar la clasificación

final. Este método demostró superar los modelos de aprendizaje profundo, especialmente en conjuntos de datos de bajo volumen y alta dimensionalidad.

Chieregato et al. (2022) desarrollaron un modelo híbrido de Machine Learning y Deep Learning para clasificar la severidad del COVID-19. Se utilizó una red neuronal convolucional como extractor de características a partir de imágenes de tomografías. Las características extraídas, junto con datos de laboratorio y clínicos, se alimentaron a un algoritmo de potenciación de gradiente CatBoost para la clasificación final. El modelo alcanzó un AUC (Área bajo la Curva) probabilístico de 0.949 en el conjunto de prueba.

Si bien el estado del arte muestra avances significativos en la clasificación de la osteoartritis a partir de radiografías mediante redes neuronales convolucionales, entre otros problemas de clasificación, la literatura reciente sugiere que los modelos de aprendizaje híbridos ofrecen un rendimiento superior.

Estos modelos combinan la potente extracción de características del Deep Learning con la eficiencia y con la interpretabilidad del Machine Learning. Ejemplos de su eficacia incluyen:

- Un modelo que integra Random Forest para extracción de características con una red LSTM para predicción de precios de acciones, logrando un RMSE de 7.13.

- Un enfoque que utiliza una CNN para la extracción de características y XGBoost como clasificador en datos de bajo volumen/alta dimensionalidad, obteniendo precisiones de hasta 95.2%.
- Un modelo para clasificar la severidad del COVID-19 que combina una CNN como extractor de características y CatBoost (un algoritmo de gradient boosting), alcanzando un AUC de 0.949.

Inspirado por esta justificada capacidad para superar los resultados de los modelos individuales, se propone la implementación de un modelo híbrido para el diagnóstico del grado de osteofito. Este modelo utilizará una CNN pre-entrenada mediante transfer learning como extractor de características automática de las radiografías (Deep Learning), y posteriormente, un clasificador de gradient boosting para la clasificación final de los grados de osteofitos (Machine Learning).

2.2. BASES TEÓRICAS

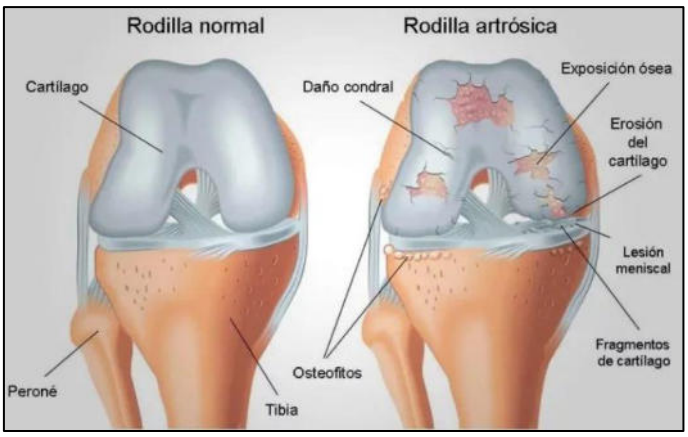
2.2.1. ANOMALÍAS OSEAS

2.2.1.1. OSTEOFITOS

> Los osteofitos se definen como la degeneración del hueso en las superficies articulares. Esto es un síntoma de osteoartritis o artrosis. (Fisioterapia Online, s.f.).

Figura 1

Comparación de rodilla sana y rodilla con osteofitos



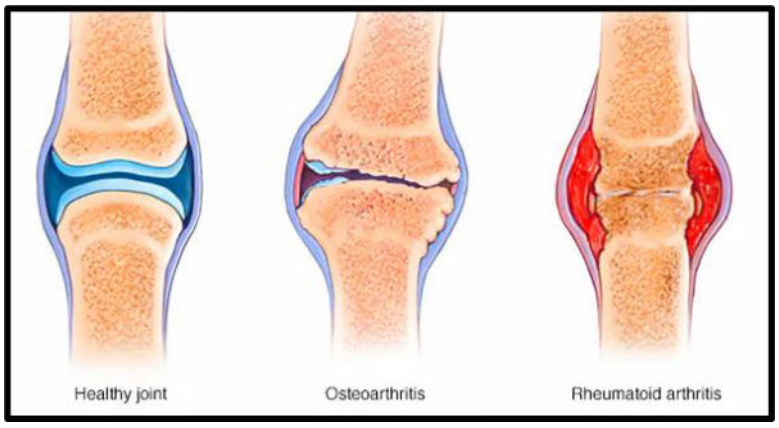
Nota. Adaptado de *Fisiolution: qué son y cómo afectan a las articulaciones del cuerpo* [Fotografía], por Fisiolution, 2023, (https://fisiolution.com/osteofitos-que-son-y-como-afectan-a-las-articulaciones-del-cuerpo/). Derechos de autor 2023 por Fisiolution Fisioterapia.

2.2.1.2. OSTEOARTRITIS

El daño articular es la causa más frecuente de los osteofitos. A medida que la osteoartritis degrada el cartílago de los extremos de los huesos, el organismo intenta reparar la pérdida generando osteofitos cerca de la zona dañada. (Mayo Clinic, 2020).

Figura 2

Comparación entre articulación sana y con osteoartritis



Nota. Adaptado de *Artritis* [Fotografía], por Mayo Clinic, 2023, (https://www.mayoclinic.org/es/diseases-conditions/arthritis/symptoms-causes/syc-20350772).

Derechos de autor 2023 por Mayo Foundation for Medical Education and Research.

2.2.1.3. DIAGNÓSTICO EN RADIOGRAFÍA

La radiología diagnóstica ayuda a los proveedores de atención médica a ver estructuras dentro del cuerpo. Los médicos que se especializan en la interpretación de estas imágenes se denominan radiólogos de diagnóstico. (Medlineplus, 2017).

Figura 3

Lectura de radiografía



Nota. Adaptado de *¿En qué consiste la radiografía?* [Fotografía], por Top Doctors España, 2023 (https://www.topdoctors.es/diccionario-medico/radiografia). Derechos de autor 2023 por Top Doctors International.

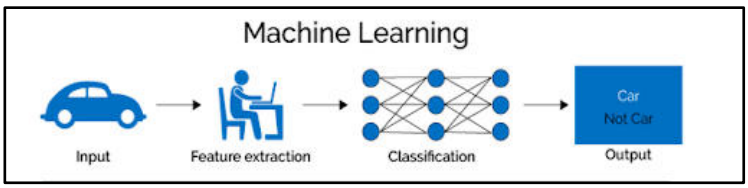
2.2.2. MACHINE LEARNING

2.2.2.1. DEFINICIÓN

Subconjunto de la informática, la ciencia de datos y la inteligencia artificial (IA) que permite a los sistemas aprender y mejorar a partir de los datos sin intervenciones adicionales de programación. En lugar de utilizar instrucciones explícitas para optimizar el rendimiento, los modelos de Machine Learning se basan en algoritmos y modelos estadísticos que implementan tareas basadas en patrones de datos e inferencias. (IBM, s.f.).

Figura 4

Ejemplo sobre Machine Learning



Nota. Adaptado de *Feature extraction is only required for ML algorithms* [Fotografía], por Built In, 2023 (https://builtin.com/machine-learning/deep-learning).

Derechos de autor 2023 por Built In.

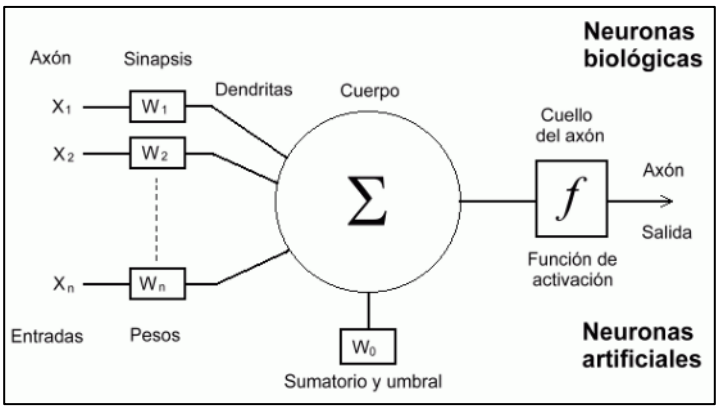
2.2.3. REDES NEURONALES ARTIFICIALES

2.2.3.1. DEFINICIÓN

Una red interconectada de nodos semejante a las neuronas en un cerebro. Cada nodo es una neurona artificial y cada flecha una conexión entre neuronas. (Schmidhuber, J., 2015)

Figura 5

Esquema de red neuronal artificial



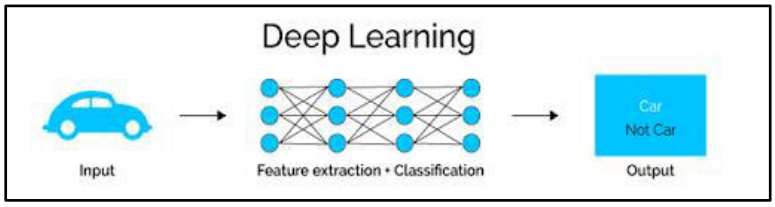
Nota. Adaptado de *Redes Neuronales: una visión superficial* [Fotografía], por Francisco Sancho, (https://www.cs.us.es/~fsancho/Blog/posts/Redes_Neuronales.md.html). Derechos de autor por Universidad de Sevilla.

2.2.3.2. DEEP LEARNING

Subconjunto de Machine Learning que utiliza redes neuronales multicapa, llamadas redes neuronales profundas, para simular el complejo poder de toma de decisiones del cerebro humano. Las redes neuronales profundas se componen de varias capas de nodos interconectados, cada una de las cuales se basa en la capa anterior para refinar y optimizar la predicción o la categorización. (IBM, s.f.)

Figura 6

Ejemplo de Deep Learning



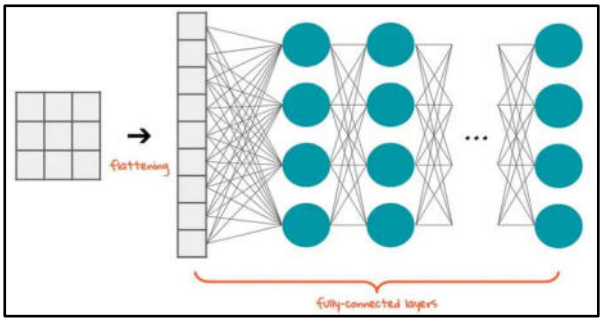
Nota. Adaptado de *Feature extraction is only required for ML algorithms* [Fotografía], por Built In, 2023 (https://builtin.com/machine-learning/deep-learning). Derechos de autor 2023 por Built In.

2.2.3.3. REDES NEURONALES CONVOLUCIONALES

Red neuronal organizada en capas que constan de una serie de nodos interconectados, que extraen características de las imágenes, donde aprenden mientras la red se entrena con una colección de imágenes. (Analytics Vidhya, 2022)

Figura 7

Diseño de una red neuronal convolucional

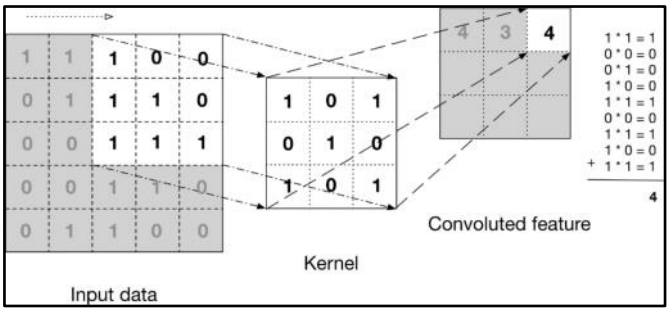


Nota. Adaptado de *Convolutional Neural Networks(CNN) Tutorial* [Fotografía], por Onur Akköse, 2020, Medium (https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-a78e78c1ba94). Derechos de autor 2020 por Medium.

La convolución es una operación matemática sencilla que se suele utilizar para el tratamiento y el reconocimiento de imágenes. En una imagen, su efecto se asimila a un filtrado. (Data Scientest, 2023)

Figura 8

Ejemplo de una operación de convolución



Nota. Adaptado de *Convolution* [Fotografía], por K. Hawkins, 2023, Medium (https://medium.com/@knighthawkk/how-to-use-conv2d-layers-as-fully-connected-layers-b0a82eb8a408). Derechos de autor 2023 por Medium.

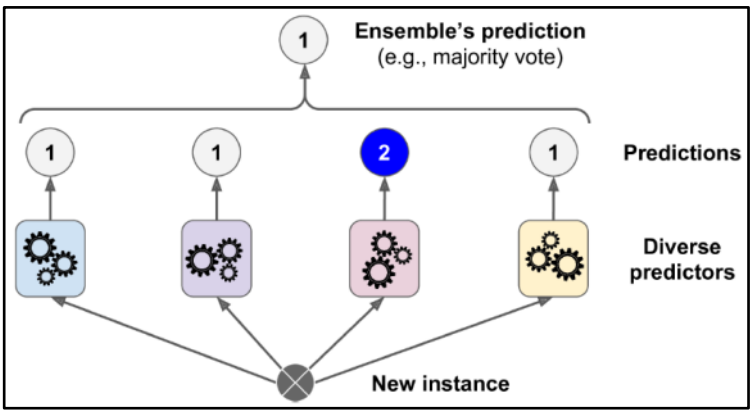
2.2.4. MÉTODOS DE ENSAMBLAJE

2.2.4.1. DEFINICIÓN

Las técnicas de Ensamble combinan modelos individuales para mejorar la estabilidad y el desempeño de la predicción del modelo de Machine Learning. Permiten una mejora en el rendimiento de la predicción combinando múltiples modelos de aprendizaje en un solo modelo, cada una de las predicciones se combina para obtener una única predicción. (Medium, 2020)

Figura 9

Ejemplo de método de ensamblaje: Voting



Nota. Adaptado de *Clasificador de votación* [Fotografía], por Samuel Cortez, 2022, Medium. (https://medium.com/@oscars.cortezmo/introducci%C3%B3n-a-los-m%C3%A9todos-de-ensamble-y-al-algoritmo-de-xgboost-caso-pr%C3%A1ctico-e8cb0d58394b). Derechos de autor 2022 por el autor.

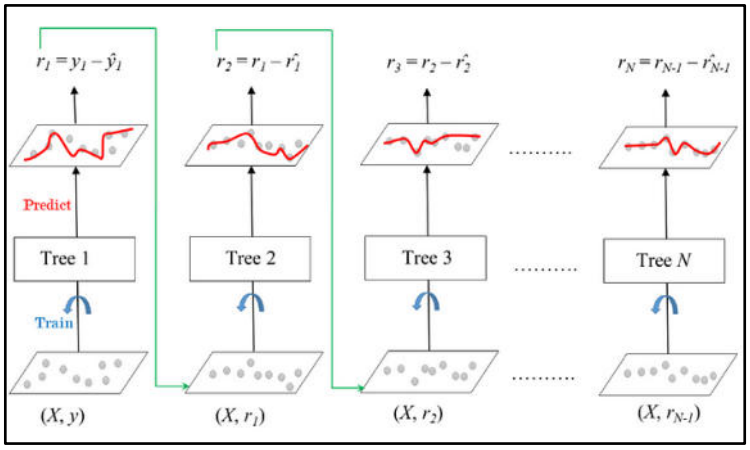
2.2.4.2. GRADIENT BOOSTING

Algoritmo popular en donde cada predictor corrige los errores de los predecesores, trabaja de forma secuencial agregando predictores a un ensamble en donde cada uno corrige a su predecesor, en lugar de ajustar los pesos de las instancias en cada iteración, este método intenta ajustar el nuevo predictor a los errores residuales cometidos por el predictor anterior. El objetivo principal del método es el de minimizar una función de pérdida agregando aprendices

débiles utilizando el algoritmo de optimización de descenso de gradiente. (Medium, 2020)

Figura 10

Ejemplo de Gradient Boosting



Nota. Adaptado de *Gradient Boosting* [Fotografía], por Samuel Cortez , 2022, Medium. (https://medium.com/@oscars.cortezmo/introducci%C3%B3n-a-los-m%C3%A9todos-de-ensamble-y-al-algoritmo-de-xgboost-caso-pr%C3%A1ctico-e8cb0d58394b). Derechos de autor 2022 por el autor.

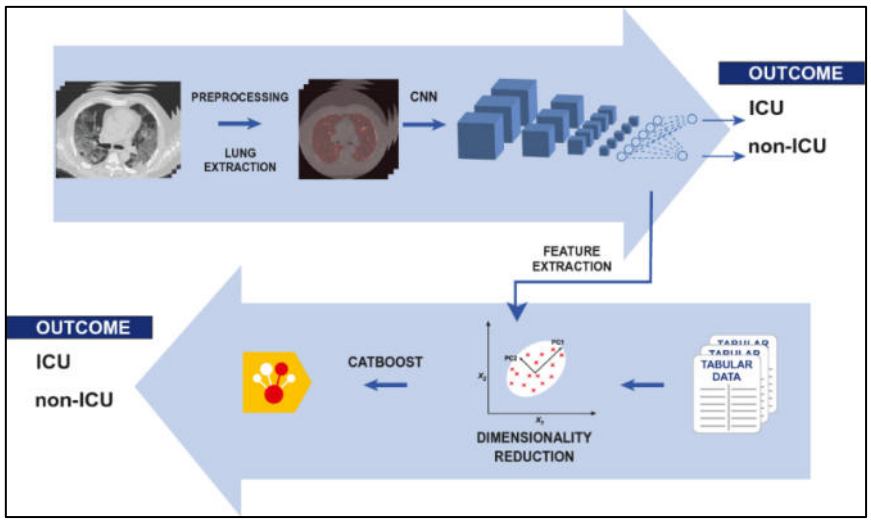
2.2.5. MODELOS HÍBRIDOS DE DEEP LEARNING Y MACHINE LEARNING

Un modelo híbrido de Deep Learning y Machine Learning es una estrategia sofisticada en el aprendizaje automático que busca combinar de forma sinérgica las capacidades de ambas disciplinas para superar las limitaciones inherentes a cada una cuando se utilizan

de forma aislada. Su objetivo principal es lograr un alto rendimiento predictivo, especialmente en escenarios desafiantes con datos de alta dimensión y bajo volumen, que son frecuentes en campos como la biología y la medicina. (Mavaie, P., 2023)

Figura 11

Enfoque de modelo híbrido de Machine Learning



Nota. Adaptado de "A hybrid machine learning/deep learning COVID-19 severity predictive model from CT images and clinical data" (Figura 1, p. 2), por Chieregato et al., 2022, *Scientific Reports*, 12 (1), Article 4329.

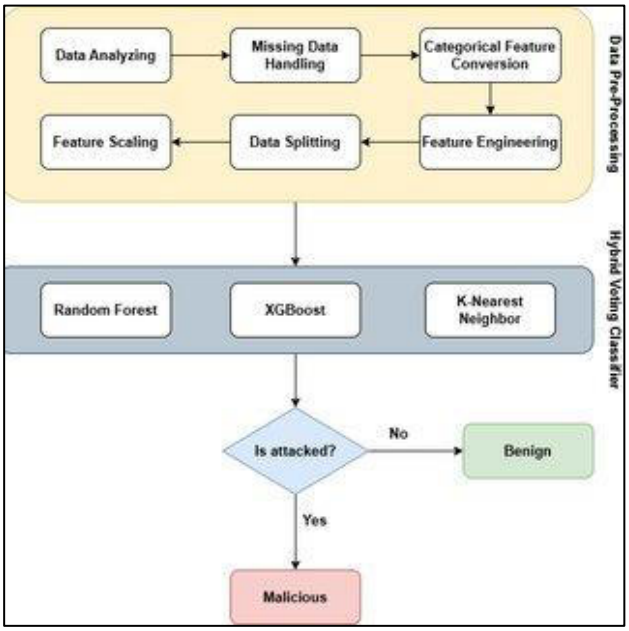
La integración entre Machine Learning y Deep Learning en un sistema híbrido puede lograrse mediante diversas estrategias, incluyendo:

- Pipelines (tuberías): Donde las técnicas tradicionales de Deep Learning pueden preprocesar los datos para extraer

características y se alimentan a modelos de Machine Learning para un procesamiento adicional.

Figura 12

Diagrama de flujo del pipeline de un modelo híbrido

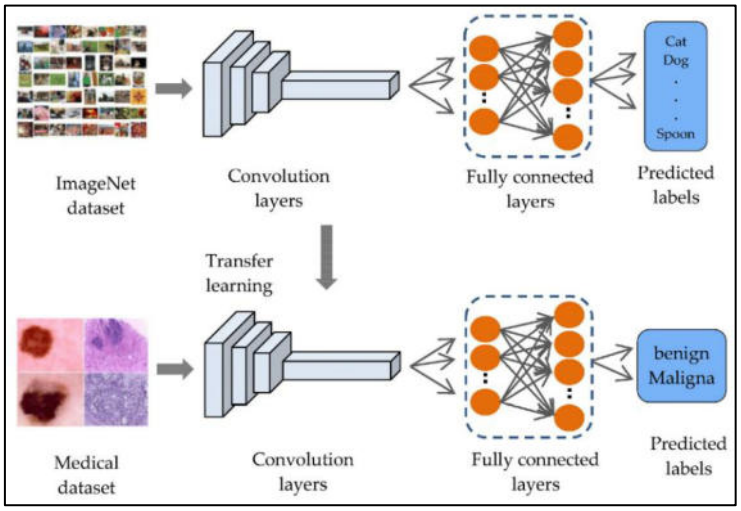


Nota. Adaptado de "Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset" (Figura 3, p. 6), por Akif et al., 2025, arXiv preprint arXiv:2502.12382.

- Transferencia de conocimiento (Transfer Learning): Utilizando el conocimiento aprendido en un dominio, comúnmente por un modelo de Deep Learning pre-entrenado en grandes conjuntos de datos, para mejorar el rendimiento en una tarea relacionada, pero con datos limitados.

Figura 13

Aplicación de Transfer Learning a ImageNet



Nota. Adaptado de "Incorporating a Novel Dual Transfer Learning Approach for Medical Images" (Figura 1, p. 2) Mukhlif et al., 2023, *Sensors*, 23(2), 570.

2.3. DEFINICIÓN DE TÉRMINOS

Sistema = Conjunto de componentes coordinados para un propósito común.

Técnica = Procedimiento específico para desarrollar una tarea.

Métrica = Medida cuantitativa para evaluar el rendimiento de un modelo.

Parámetro = Variable interna del modelo cuyo valor se aprende durante el entrenamiento o función.

Algoritmo = Conjunto de instrucciones encargadas de resolver un problema.

Arquitectura = Diseño estructural de un modelo.

Inteligencia Artificial = Campo que desarrolla sistemas capaces de realizar tareas que requieran inteligencia humana.

Grado Kellgren-Lawrence = Sistema de clasificación para el diagnóstico temprano de osteofitos en radiografías.

Transfer Learning = Técnica que permite adaptar un modelo entrenado para una tarea específica.

EfficientNet = Familia de arquitecturas CNN escaladas de manera equilibrada en profundidad, anchura y resolución para optimizar precisión y eficiencia computacional.

ResNet = Arquitectura CNN que introduce conexiones residuales para facilitar el entrenamiento de redes muy profundas.

DenseNet = Arquitectura CNN donde cada capa está conectada a todas las capas posteriores, promoviendo la reutilización de características.

Inception = Arquitectura CNN que utiliza convoluciones paralelas de diferentes tamaños para capturar patrones a múltiples escalas.

Epochs = Un ciclo completo de entrenamiento en el que el modelo ha visto y aprendido de todas las imágenes del conjunto de entrenamiento una vez.

Learning Rate = Parámetro que regula los ajustes de conexiones internas en cada paso del entrenamiento del modelo.

Loss Function = Función matemática que mide el error de la predicción del modelo en comparación con la etiqueta real.

Dropout = Técnica de regularización que apaga aleatoriamente neuronas durante el entrenamiento para prevenir sobreajuste.

Data Augmentation = Técnica de datos que nos permite aumentar nuestro dataset de dos formas: Introduciendo perturbaciones en los datos originales o utilizando distintas distribuciones.

Overfitting = Situación donde el modelo no puede generalizar datos nuevos, debido a la memorización de los datos de entrenamiento.

Regularizador L1/L2 = Técnicas que penalizan pesos grandes añadiendo términos a la función de pérdida (L1: promueve dispersión, L2: evita valores extremos).

Label Smoothing = Técnica de regularización que suaviza las etiquetas one-hot para mejorar la generalización del modelo.

Desbalance de clases = Situación en la que un conjunto de datos tiene muchas más muestras de una clase que de otras.

Train/Val/Test = Conjuntos de datos para entrenar el modelo, validar y probar el rendimiento final del modelo.

CNN = Red neuronal diseñada para imágenes usando convoluciones con el fin de detectar bordes, texturas y formas.

Fine-Tuning = Proceso de reentrenar una red pre-entrenada en un nuevo dataset, ajustando capas finales (o todas) para una tarea específica.

Gradient Boosting = Algoritmo que construye árboles de decisión de forma secuencial que corrige los errores del anterior.

XGBoost = Algoritmo de gradient boosting basado en árboles que optimiza el gradiente e incluye regularización para alto rendimiento.

LightGBM = Algoritmo de gradient boosting basado en histogramas y crecimiento vertical de árboles.

CatBoost = Algoritmo de boosting especializado en manejar variables categóricas nativamente sin necesidad de preprocesamiento extenso.

Modelo Híbrido = Arquitectura que combina dos tipos de modelos, CNN para la percepción visual y GB para la clasificación final.

Clasificación Jerárquica = La estrategia de descomponer un problema de clasificación complejo en una serie de problemas más simples y organizados en niveles.

SMOTE = Técnica para aumentar la clase minoritaria generando ejemplos sintéticos en el espacio de características.

Matriz de Confusión = Tabla que desglosa el rendimiento de un modelo, mostrando cuántas veces acertó o confundió una clase con otra.

Precisión = Métrica de las veces que el modelo predijo una clase. Midiendo el porcentaje de la calidad de las predicciones.

Recall = Métrica de las veces que las muestras reales de una clase. Midiendo el porcentaje que logró no pasar por alto los casos.

F1-Score = Métrica que balancea el rendimiento general en datasets desbalanceados.

Accuracy = Métrica que mide la proporción de predicciones correctas sobre el total de muestras.

3. METODOLOGÍA

3.1. TIPO Y NIVEL DE INVESTIGACIÓN

- Tipo de investigación: Investigación aplicada.
- Nivel de investigación: Investigación original.

La investigación aplicada es un tipo de investigación que se realiza para resolver un problema práctico. Paitán et al. (2019) afirmaron lo siguiente:

> Se llaman aplicadas porque se basan en los resultados de la investigación básica, pura o fundamental, de las ciencias naturales y sociales, que hemos visto, se formulan problemas e hipótesis de trabajo para resolver los problemas de la vida social de la comunidad regional o del país. (p. 136)

La investigación original es un tipo de investigación que genera nuevos conocimientos. En esta investigación, el nuevo conocimiento generado es un nuevo método para diagnosticar el grado de osteofito en radiografías utilizando un modelo híbrido de Machine Learning.

3.2. ÁMBITO TEMPORAL Y ESPACIAL

El presente trabajo de investigación se realizó en la región de Moquegua con datos proporcionados por el repositorio de Kaggle y Osteoarthritis Initiative (OAI).

3.3. POBLACIÓN Y MUESTRA

3.3.1. POBLACIÓN

La población motivo de esta investigación está conformada por 8260 imágenes de radiografías del dataset brindado por Kaggle y Osteoarthritis Initiative (OAI).

3.3.2. MUESTRA

La muestra de esta investigación está conformada por el total de imágenes de la población mencionada, donde se encuentra dividida en conjuntos para entrenamiento, validación y prueba; todos los conjuntos contienen 5 clases.

3.4. INSTRUMENTOS

3.4.1. INSTRUMENTOS DE CÓMPUTO

3.4.1.1. HARDWARE

Para el entrenamiento de los modelos desarrollados, fue suficiente contar con un ordenador portátil de características básicas, equipado con una GPU o CPU. Si bien estos componentes permiten acelerar el procesamiento, el enfoque metodológico adoptado no requirió una infraestructura computacional especializada, lo que demuestra la viabilidad del desarrollo en entornos accesibles.

3.4.1.2. SOFTWARE

Se utilizó el lenguaje de programación Python 3.9 como entorno de desarrollo, debido a su extensa variedad de librerías dedicadas a Machine Learning. Se empleó Jupyter Notebook en el entorno de trabajo mediante la distribución Anaconda, debido a su habilidad para unir código, texto, etc.

Como framework principal de Deep Learning se optó por:

- TensorFlow (v2.10). Un framework de bajo nivel para los cálculos numéricos y la ejecución automática en GPU.

- Keras (v2.10): La API de alto nivel sobre TensorFlow utilizada para construir, entrenar y evaluar las arquitecturas de Redes Neuronales Convolucionales (CNN).

Como librerías de Machine Learning y procesamiento de datos se utilizó:

- Scikit-learn: Utilizada para métricas de evaluación (classification_report, accuracy_score, confusion_matrix).
- XGBoost, LightGBM, CatBoost: Las tres librerías de Gradient Boosting implementadas como los clasificadores finales en la arquitectura híbrida.
- Imbalanced-learn: Utilizada específicamente para la técnica de sobremuestreo SMOTE, para corregir el desbalance de clases en los vectores de características.
- NumPy: La librería fundamental para el manejo de arrays y operaciones matemáticas de alto rendimiento.
- Pandas: Utilizada para la manipulación inicial de los datos y la gestión de las rutas de los archivos de imagen.

Como librerías de visualización:

- Matplotlib y Seaborn: Utilizadas para la visualización de datos, incluyendo las curvas de aprendizaje (pérdida y precisión) y las matrices de confusión.

3.4.2. INSTRUMENTOS CONCEPTUALES Y TÉCNICAS

3.4.2.1. CLASIFICACIÓN JERÁRQUICA

Esta estrategia aborda problemas de clasificación complejos basándose en el principio de "divide y vencerás". En lugar de forzar a un solo modelo a resolver el problema de clasificar 5 clases, donde algunas se parecen entre sí o son escasas, se divide en problemas más sencillos y manejables, organizados en una estructura de jerarquía (Silla & Freitas, 2011).

3.4.2.2. TRANSFERN LEARNING

Para esta investigación se implicó tomar una red neuronal convolucional de última generación pre-entrenada con millones de imágenes (ImageNet) para extraer características relevantes de las radiografías. Esto aprovecha el conocimiento del modelo que tiene sobre formas, texturas, patrones.

3.4.2.3. MODELADO HÍBRIDO

Esta combinación de Machine Learning y Deep Learning permite que la CNN actúe como un "par de ojos" que extrae información compleja de la radiografía, y luego con un clasificador de Gradient Boosting actúe como un "cerebro" que toma la decisión final basándose en esa información.

3.4.2.4. MANEJO DE DESBALANCE

Como instrumentos que se aplicaron para combatir el desbalance de clases, se optó por el cálculo de pesos para la atención de clases minoritarias y SMOTE como sintetizador de muestras nuevas para nivelar la cantidad entre las clases.

3.4.2.5. TÉCNICA DE ENSAMBLE VOTING

Esta técnica aprovecha los mejores modelos entrenados, en lugar de confiar en un único modelo, se crea un comité donde cada modelo emite una predicción y luego se promedian las probabilidades de cada uno.

3.4.2.6. ECUACIONES DE MÉTRICAS DE EVALUACIÓN

Se usaron métricas estándar de clasificación basadas en ecuaciones derivadas de la matriz de confusión para evaluar el rendimiento de los modelos.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1Score = 2 \cdot \frac{precision \times recall}{precision + recall}$$

Donde:

- T_p (Verdaderos Positivos): Son casos correctamente predichos como positivos.
- T_n (Verdaderos Negativos): Son casos correctamente predichos como negativos.
- F_p (Falsos Positivos): Son casos incorrectamente predichos como positivos.
- F_n (Falsos Negativos): Son casos incorrectamente predichos como negativos.

3.5. PROCEDIMIENTOS

- Recolección y estructuración del dataset
- Diseño de la clasificación jerárquica.
- Pre-procesamiento de datos.
- Diseño y optimización del extractor de características.
- Entrenamiento y evaluación de clasificadores.
- Desarrollo del ensamble de modelos.
- Análisis de robustez y estabilidad del modelo propuesto.
- Diseño del modelo de predicción final.

3.6. ANÁLISIS DE DATOS

Para el desarrollo de la investigación, se detallará el proceso metodológico siguiente para la implementación y evaluación del modelo hibrido.

3.6.1. RECOLECCIÓN Y ESTRUCTURACIÓN DEL DATASET

3.6.1.1. FUENTE Y CONTENIDO

El estudio utilizó el dataset público "Knee Osteoarthritis Dataset with Severity", disponible en la plataforma Kaggle. Este conjunto de datos contiene radiografías de rodillas con osteofitos clasificadas por expertos según la escala de Kellgren-Lawrence (KL) en 5 grados: Sano, Dudoso, Mínimo, Moderado y Severo.

Tabla 1

Cantidad de imágenes del Dataset

Clase	Train	Val	Test
Grado 0: Sano	2286	328	639
Grado 1: Dudoso	1046	153	296
Grado 2: Mínimo	1516	212	447
Grado 3: Moderado	757	106	223
Grado 4: Severo	173	27	51

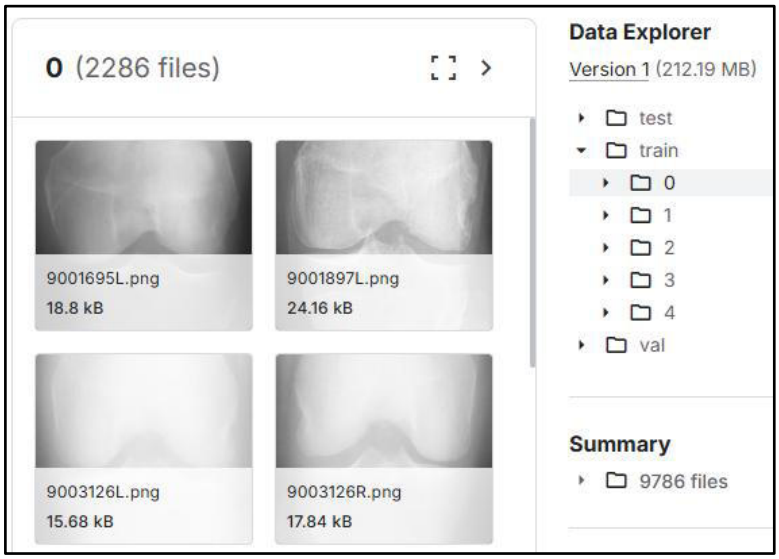
Nota. Las clases representan el grado de osteofitos y los datos se dividen en conjuntos de entrenamiento, validación y prueba.

3.6.1.2. PARTICIÓN DE DATOS

Se adoptó la partición predefinida del dataset, la cual divide las imágenes en tres conjuntos mutuamente excluyentes: entrenamiento, validación y prueba (Train, Val, Test). Esta estructura fija se mantuvo a lo largo de todos los experimentos para garantizar la consistencia y la comparabilidad de los resultados.

Figura 14

Exploración del dataset



Nota. Adaptado de *Knee Osteoarthritis Dataset with Severity*, por Shashwat, 2022, Kaggle (https://www.kaggle.com/datasets/shashwatwork/knee-osteoarthritis-dataset-with-severity/data). Derechos de autor 2022 por Kaggle.

3.6.2. DISEÑO DE LA CLASIFICACIÓN JERÁRQUICA

3.6.2.1. CLASIFICACIÓN JERARQUICA

El análisis exploratorio inicial confirmó un severo desbalance de clases y una alta similitud visual entre los grados adyacentes, lo que representa un desafío significativo para un solo clasificador. Para abordar esta complejidad, se diseñó esta estrategia donde el problema de 5 clases se descompuso en dos subproblemas más sencillos y clínicamente relevantes:

- Modelo 1 (Clasificador Binario): Especializado en la tarea de alto nivel de diferenciar casos "No Severos" (Grados 0 y 1) de casos "Severos" (Grados 2, 3 y 4).
- Modelo 2 (Clasificador Severo): Especializado en la tarea más fina de clasificar únicamente entre los grados 2, 3 y 4, activándose solo si el Modelo 1 identifica un caso como "Severo".

Tabla 2

Distribución de imágenes del Modelo Binario

Clase	Train	Val	Test
No severo	2286+1046	328+153	639 + 296
Severo	1516+757+173	212+106+27	447+223+51

Nota. Se agrupó los grados 0 y 1 como una clase No severa y los grados 2, 3, 4 como una clase Severa.

Tabla 3

Distribución de imágenes del Modelo Severo

Clase	Train	Val	Test
Grado 2: Mínimo	1516	212	447
Grado 3: Moderado	757	106	223
Grado 4: Severo	173	27	51

Nota. Se mantuvieron los grados con presencia de osteofitos para el modelo severo.

3.6.3. PRE-PROCESAMIENTO DE DATOS

3.6.3.1. PRE-PROCESAMIENTO DEL DATASET

El framework TensorFlow se encargó del preprocesamiento automatizado de las imágenes, las cuales mantuvieron su resolución nativa de 224x224 píxeles. Donde la función image_dataset_from_directory() se configuró con parámetros específicos: batch_size=32 para el tamaño de lote, shuffle=True exclusivamente para el conjunto de entrenamiento para evitar sobreajuste a patrones de orden, y label_mode='categorical' para generar representaciones one-hot de las etiquetas de las clases. Posteriormente, se aplicaron técnicas de optimización mediante cache() y prefetch() para almacenamiento en memoria de los datos preprocesados y con tamaño de buffer automático Autotune para paralelizar las operaciones de computación, reduciendo significativamente los tiempos de entrenamiento.

Figura 15

Código de carga y optimización del dataset

```
# Creación de carga para train, val, test
train_dataset = tf.keras.utils.image_dataset_from_directory(
    TRAIN_DIR, labels='inferred', label_mode='categorical',
    image_size=(IMG_SIZE, IMG_SIZE), batch_size=BATCH_SIZE, shuffle=True
)
validation_dataset = tf.keras.utils.image_dataset_from_directory(
    VAL_DIR, labels='inferred', label_mode='categorical',
    image_size=(IMG_SIZE, IMG_SIZE), batch_size=BATCH_SIZE, shuffle=False
)
test_dataset = tf.keras.utils.image_dataset_from_directory(
    TEST_DIR, labels='inferred', label_mode='categorical',
    image_size=(IMG_SIZE, IMG_SIZE), batch_size=BATCH_SIZE, shuffle=False
)
class_names = train_dataset.class_names
print(f"Clases encontradas: {class_names}")

# Optimización del dataset
AUTOTUNE = tf.data.AUTOTUNE
train_dataset = train_dataset.cache().prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.cache().prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.cache().prefetch(buffer_size=AUTOTUNE)
```

```
Found 5778 files belonging to 2 classes.
Found 826 files belonging to 2 classes.
Found 1656 files belonging to 2 classes.
Clases encontradas: ['no_severo', 'severo']
```

Nota. Código en Python que muestra la carga de imágenes para entrenamiento, validación y prueba, así como la optimización del rendimiento.

3.6.3.2. CÁLCULO DE PESOS

Dado el desbalanceo significativo entre las clases, se calculó el peso de cada clase para la función de pérdida. Se usó la función compute_class_weight() de la librería scikit-learn con la estrategia 'balanced'. Para obtener las etiquetas necesarias para el cálculo, se extrajeron y concatenaron las etiquetas one-hot del train_dataset y luego se convirtieron a su formato entero. Este enfoque obliga al modelo a asignar una mayor penalización a los errores cometidos en la clase

minoritaria durante el entrenamiento, equilibrando así la influencia de cada clase en el proceso de aprendizaje.

Figura 16

Código de cálculo de pesos de clases

```
# Cálculo de Pesos balanceados
train_labels_one_hot = np.concatenate([y for x, y in train_dataset], axis=0)
train_labels_int = np.argmax(train_labels_one_hot, axis=1)
class_weights_calculated = class_weight.compute_class_weight(
    'balanced', classes=np.unique(train_labels_int), y=train_labels_int
)
class_weight_dict = dict(enumerate(class_weights_calculated))
print(f"Pesos binarios calculados: {class_weight_dict}")
```

Nota. Código en Python que calcula los pesos para compensar el desbalance en la distribución de clases.

3.6.3.3. AUMENTACIÓN DE DATOS

Para mejorar la generalización y mitigar el sobreajuste, se aplicó una estrategia de aumentación de datos en tiempo real "on-the-fly" al conjunto de entrenamiento. Donde se implementó como una serie de capas Keras al inicio de la arquitectura de la CNN, aplicando transformaciones aleatorias como volteos horizontales (RandomFlip), rotaciones (RandomRotation), zooms (RandomZoom), ajustes de contraste (RandomContrast).

Figura 17

Código de data augmentation

```
data_augmentation = models.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.15),
    layers.RandomZoom(0.15),
    layers.RandomContrast(0.2)
], name="data_augmentation")
```

Nota. Código en Python que crea una capa de data augmentation con transformaciones aleatorias a las imágenes.

Asimismo, se priorizó la eficiencia computacional, evitando cualquier función que demandara almacenamiento, recursos y tiempo de cálculo adicionales de forma sustancial. Por ello se optó deliberadamente por no utilizar una generación de imágenes offline. Esto no contribuye a la variabilidad de los datos en cada época de entrenamiento, lo que puede limitar la capacidad de generalización del modelo.

3.6.4. DISEÑO Y OPTIMIZACIÓN DEL EXTRACTOR DE CARACTERÍSTICAS

3.6.4.1. ANÁLISIS COMPARATIVO Y SELECCIÓN DEL MODELO BASE

Se realizó un estudio comparativo de varias arquitecturas de CNN del estado del arte, incluyendo variantes de EfficientNet, ResNet y DenseNet, para seleccionar el

extractor de características más adecuado. Durante esta fase, se observó que los modelos de mayor complejidad (ej. EfficientNetB4 y superiores), si bien teóricamente más potentes, presentaban dos inconvenientes principales para este proyecto: un alto costo computacional y una fuerte tendencia al sobreajuste (overfitting) dado el tamaño del conjunto de datos.

Tabla 4

Comparación de Modelos Base para entrenamiento

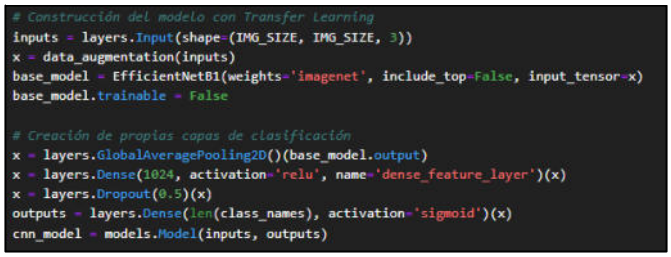
Modelo	Parámetros (Complejidad)
EfficienNetB0	5.3 M
EfficienNetB1	7.8 M
EfficienNetB2	9.2 M
EfficienNetB3	12 M
ResNet50	23.5 M
DenseNet121	6.9 M
InceptionV3	21.7 M

Nota. M = millones. Tabla elaborada por el autor con datos adaptados de los estudios de Yang et al. (2021) y Viso AI (2023).

En consecuencia, se tomó la decisión estratégica de optar por un modelo más simple, EfficientNetB0. Esta elección se fundamenta en la búsqueda de un balance óptimo entre la capacidad de representación del modelo y la eficiencia computacional, priorizando una arquitectura que pudiera ser controlada eficazmente mediante técnicas de regularización.

Figura 18

Código del modelamiento



```python
# Construcción del modelo con Transfer Learning
inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3))
x = data_augmentation(inputs)
base_model = EfficientNetB1(weights='imagenet', include_top=False, input_tensor=x)
base_model.trainable = False

# Creación de propias capas de clasificación
x = layers.GlobalAveragePooling2D()(base_model.output)
x = layers.Dense(1024, activation='relu', name='dense_feature_layer')(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(len(class_names), activation='sigmoid')(x)
cnn_model = models.Model(inputs, outputs)
```

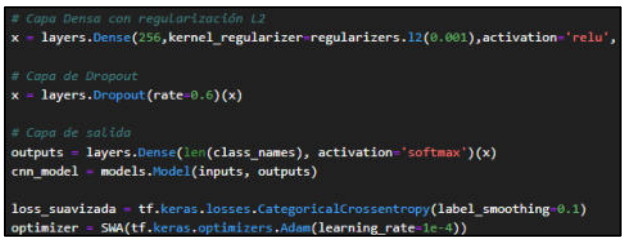
Nota. Código en Python sobre la construcción del modelo binario usando Transfer Learning con EfficientNetB1, seguido de capas personalizadas.

3.6.4.2. IMPLEMENTACIÓN DE ESTRATEGIAS DE REGULARIZACIÓN

Para mitigar el riesgo de sobreajuste inherente al modelo, se implementó un robusto esquema de regularización que incluyó: Elevar moderadamente la aumentación de datos y la tasa de Dropout, adicionar regularización como una capa Dense menor, regularizadores L2 y uso de Label Smoothing.

Figura 19

Código de regularización



```python
# Capa Densa con regularización L2
x = layers.Dense(256,kernel_regularizer=regularizers.l2(0.001),activation='relu',

# Capa de Dropout
x = layers.Dropout(rate=0.6)(x)

# Capa de salida
outputs = layers.Dense(len(class_names), activation='softmax')(x)
cnn_model = models.Model(inputs, outputs)

loss_suavizada = tf.keras.losses.CategoricalCrossentropy(label_smoothing=0.1)
optimizer = SWA(tf.keras.optimizers.Adam(learning_rate=1e-4))
```

Nota. Código en Python adicionando regularizadores para combatir el sobreajuste.

3.6.4.3. OPTIMIZACIÓN DE HIPERPARÁMETROS

Se utilizó la librería KerasTuner con la estrategia RandomSearch para encontrar la configuración óptima de hiperparámetros de la "cabeza" de clasificación de la CNN, ajustando la tasa de aprendizaje, el número de neuronas y la tasa de Dropout.

Figura 20

Código de búsqueda de hiperparámetros ideales

```
tuner = kt.RandomSearch(
    build_hypermodel,
    objective='val_accuracy',
    max_trials=10,
    executions_per_trial=2
)
tuner.results_summary()
stop_early = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)
print("\n--- INICIANDO BÚSQUEDA DE HIPERPARÁMETROS ---")
tuner.search(
    train_dataset,
    epochs=20,
    validation_data=validation_dataset,
    callbacks=[stop_early],
    class_weight=class_weight_dict
)
print("\n--- BÚSQUEDA FINALIZADA ---")
```

```
Trial 10 Complete [00h 55m 55s]
val_accuracy: 0.7518159747123718

Best val_accuracy So Far: 0.7554479241371155
Total elapsed time: 11h 53m 11s
INFO:tensorflow:Oracle triggered exit

--- BÚSQUEDA FINALIZADA ---
```

```
best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]
print(f"""
La búsqueda ha finalizado. Los hiperparámetros óptimos son:
- Neuronas en la capa densa: {best_hps.get('units')}
- Tasa de Dropout: {best_hps.get('dropout')}
- Tasa de Aprendizaje: {best_hps.get('learning_rate')}
""")
best_cnn_model = tuner.get_best_models(num_models=1)[0]
```

```
La búsqueda ha finalizado. Los hiperparámetros óptimos son:
- Neuronas en la capa densa: 1024
- Tasa de Dropout: 0.4
- Tasa de Aprendizaje: 0.0001
```

Nota. Código en Python sobre el proceso de búsqueda de hiperparámetros mediante RandomSearch de Keras Tuner.

3.6.4.4. CALLBACKS

En el proceso de entrenamiento, se integraron funciones asíncronas con el propósito de aumentar la eficiencia del modelo. Estas funciones detienen automáticamente el proceso cuando no se observa mejora, monitoreando en el valor de la pérdida de validación, lo cual indicaría que el modelo ha llegado a su meseta de aprendizaje. Adicionalmente, ajustan dinámicamente la tasa de aprendizaje multiplicando el valor actual por un factor ingresado, con el fin de prevenir estancamientos durante el entrenamiento. Ambas funciones son propias de la librería de Keras.

Figura 21

Código de callbacks

```
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(
    monitor='val_loss', factor=0.2, patience=3, verbose=1
)
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss', patience=14, restore_best_weights=True, verbose=1
)
```

Nota. Código en Python sobre la creación de callbacks para reducir la tasa de aprendizaje y la detención temprana del entrenamiento.

3.6.4.5. FINE-TUNING

Se implementó una estrategia de entrenamiento de dos etapas. Primero, se entrenó únicamente la cabeza de clasificación con el modelo base congelado. Posteriormente, se descongelaron las capas de un bloque convolucional del modelo base y se continuó el entrenamiento con una tasa de aprendizaje muy baja para especializar el extractor en el dominio de las radiografías.

Figura 22

Código de reentrenamiento con Fine-Tuning

```
base_model.trainable = True
fine_tune_at = 108

for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False

cnn_model.compile(
    optimizer=optimizer_ft,
    loss=loss_suavizada,
    metrics=['accuracy']
)
```

```
history_fine = cnn_model.fit(
    train_dataset,
    epochs=total_epochs,
    initial_epoch=history_cnn.epoch[-1]+1,
    validation_data=validation_dataset,
    callbacks=[reduce_lr, early_stopping],
    class_weight=class_weight_dict
)
```

Nota. Código en Python sobre el re-entrenamiento del modelo con Fine-Tuning y configuración de capas descongeladas.

3.6.5. ENTRENAMIENTO Y EVALUACIÓN DE CLASIFICADORES

3.6.5.1. EXTRACCIÓN DE CARACTERÍSTICAS

La CNN optimizada y entrenada en la etapa anterior se utilizó como un extractor de características no supervisado. Para ello, se redefinió la arquitectura del modelo, tomando como entrada la capa original (cnn_model.input) y como salida la activación de la capa densa oculta previamente diseñada para representar características de alto nivel ('dense_feature_layer'). Esta capa produce un vector denso y compacto que encapsula la información semántica más relevante de cada imagen. Paralelamente, se extrajeron las etiquetas de los conjuntos de validación y prueba, convirtiendo las representaciones one-hot encoding de nuevo a etiquetas enteras para su uso posterior en la etapa de clasificación con el modelo híbrido.

Figura 23

Código del extractor de características

```
# Creación del Modelo Extractor
feature_extractor_model = models.Model(
    inputs=cnn_model.input,
    outputs=cnn_model.get_layer('dense_feature_layer').output
)

# Extracción de Vectores de Características
train_features = feature_extractor_model.predict(train_dataset, verbose=1)
val_features = feature_extractor_model.predict(validation_dataset, verbose=1)
test_features = feature_extractor_model.predict(test_dataset, verbose=1)

# Creación de etiquetas
val_labels_int = np.argmax(np.concatenate([y for x, y in validation_dataset], axis=0), axis=1)
test_labels_int = np.argmax(np.concatenate([y for x, y in test_dataset], axis=0), axis=1)
```

Nota. Código en Python sobre el extractor y etiquetas del conjunto de validación y prueba.

3.6.5.2. MANEJO DE DESBALANCE

Sobre los vectores de características y sus etiquetas creadas del entrenamiento, se aplicó la técnica de sobremuestreo SMOTE (Synthetic Minority Over-sampling Technique) para crear un conjunto de datos con clases minoritarias perfectamente balanceado para el clasificador final.

Figura 24

Código de sobremuestreo

```
from imblearn.over_sampling import SMOTE

print("Distribución original:", np.bincount(train_labels_int))
smote = SMOTE(random_state=42)
features_resampled, labels_resampled = smote.fit_resample(train_features, train_labels_int)

print("Nueva distribución:", np.bincount(labels_resampled))
```

Nota. Código en Python sobre el sobremuestreo del nuevo conjunto de entrenamiento para las clases minoritarias.

3.6.5.3. EVALUACIÓN DE CLASIFICADORES

Para la etapa de clasificación dentro del modelo híbrido, se seleccionaron e implementaron tres variantes modernas del algoritmo de Gradient Boosting (XGBoost, LightGBM y CatBoost), reconocidas por su alto rendimiento, eficiencia computacional y capacidad para manejar datos tabulares. Se usaron estrategias de regularización para mejorar la generalización y evitar el sobreajuste, se aplicó submuestreo

para entrenar cada árbol con subconjuntos aleatorios de datos y características, aportando diversidad y robustez.

Figura 25

Código del clasificador XGBoost

```
xgb_classifier = xgb.XGBClassifier(
    objective='multi:softmax',
    num_class=len(class_names),
    eval_metric='mlogloss',
    use_label_encoder=False,
    n_estimators=2000,
    max_depth=6,
    learning_rate=0.01,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_lambda=1,
    reg_alpha=0.1,
    gamma=0.1
)
```

Nota. Código en Python sobre la construcción del clasificador XGBoost y sus hiperparámetros.

Figura 26

Código del clasificador LightGBM

```
lgb_classifier = lgb.LGBMClassifier(
    objective='multiclass',
    num_class=3,
    n_estimators=2000,
    learning_rate=0.01,
    num_leaves=40,
    max_depth=-1,
    min_child_samples=20,
    reg_alpha=0.1,
    reg_lambda=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    verbose=-1
)
```

Nota. Código en Python sobre la construcción del clasificador LightGBM y sus hiperparámetros.

Figura 27

Código del clasificador CatBoost

```
cat_classifier = cb.CatBoostClassifier(
    objective='MultiClass',
    iterations=2000,
    learning_rate=0.02,
    depth=7,
    l2_leaf_reg=4,
    border_count=128,
    random_strength=0.5,
    bagging_temperature=0.8,
    verbose=0
```

Nota. Código en Python sobre la construcción del clasificador CatBoost y sus hiperparámetros

La evaluación de los tres clasificadores se realizó de forma exhaustiva sobre el conjunto de prueba, mediante un reporte de clasificación detallado que incluyó Precision, Recall y F1-Score para cada clase, además de una matriz de confusión para visualizar los aciertos y errores entre clases.

Figura 28

Código de reporte de clasificación y matriz de confusión

```
y_pred_xgb = xgb_classifier.predict(test_features)
print(classification_report(test_labels_int, y_pred_xgb, target_names=class_names))

cm = confusion_matrix(test_labels_int, y_pred_xgb)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicción (XGBoost)')
plt.ylabel('Etiqueta Real')
plt.title('Matriz de Confusión - Modelo de Casos Severos')
plt.show()
```

Nota. Código en Python sobre la evaluación gráfica del rendimiento de la clasificación.

Finalmente, la exactitud (accuracy) global de cada modelo en el conjunto de prueba, calculado con la función sklearn.metrics.accuracy_score, sirvió como métrica principal para comparar su desempeño y seleccionar el clasificador óptimo para la tarea.

Figura 29

Código de evaluación de exactitud

```
from sklearn.metrics import accuracy_score

acc_xgb = accuracy_score(test_labels_int, y_pred_xgb)
acc_lgb = accuracy_score(test_labels_int, y_pred_lgb)
acc_cat = accuracy_score(test_labels_int, y_pred_cat)

print(f"Precisión de XGBoost:   {acc_xgb:.2f}")
print(f"Precisión de LightGBM:  {acc_lgb:.2f}")
print(f"Precisión de CatBoost:  {acc_cat:.2f}")
```

Nota. Código en Python sobre el cálculo de exactitud para cada clasificador de Gradient Boosting.

3.6.6. DESARROLLO DEL ENSAMBLE DE MODELOS

Para maximizar la robustez y precisión, se desarrolló un modelo de ensamble a partir de los tres clasificadores de Gradient Boosting entrenados en el paso anterior. Se implementó una estrategia de votación suave (soft voting). Este método consiste en promediar las probabilidades de predicción de cada uno de los tres modelos y seleccionar la clase con la probabilidad promedio más alta como la predicción final del sistema.

Figura 30

Código del ensamble de modelos binario y severo



```
probs_xgb_binario = xgb_binario.predict_proba(features_binarias)
probs_lgb_binario = lgb_binario.predict_proba(features_binarias)
probs_cat_binario = cat_binario.predict_proba(features_binarias)

prediccion_xgb_1 = class_names_binario[np.argmax(probs_xgb_binario, axis=1)[0]]
prediccion_lgb_1 = class_names_binario[np.argmax(probs_lgb_binario, axis=1)[0]]
prediccion_cat_1 = class_names_binario[np.argmax(probs_cat_binario, axis=1)[0]]

print("\nResultados de los clasificadores:")
print(f"XGBoost: '{prediccion_xgb_1}'")
print(f"LightGBM: '{prediccion_lgb_1}'")
print(f"CatBoost: '{prediccion_cat_1}'")

promedio_probs_binario = (probs_xgb_binario + probs_lgb_binario + probs_cat_binario) / 3.0
prediccion_voting_binario = np.argmax(promedio_probs_binario, axis=1)[0]
prediccion_binario = class_names_binario[prediccion_voting_binario]

print(f"Resultado Modelo 1: La imagen es clasificada como '{prediccion_binario}'.")

if prediccion_binario == 'no_severo':
    return "\nDiagnóstico: No Severo (Grado 0 o 1)"
else:
    print("\nEl caso parece severo. Pasando al Modelo 2 para especificar el grado...")

    features_severas = extractor_severo.predict(img_array, verbose=0)

    probs_xgb_severo = xgb_severo.predict_proba(features_severas)
    probs_lgb_severo = lgb_severo.predict_proba(features_severas)
    probs_cat_severo = cat_severo.predict_proba(features_severas)

    prediccion_xgb_2 = class_names_severo[np.argmax(probs_xgb_severo, axis=1)[0]]
    prediccion_lgb_2 = class_names_severo[np.argmax(probs_lgb_severo, axis=1)[0]]
    prediccion_cat_2 = class_names_severo[np.argmax(probs_cat_severo, axis=1)[0]]

    print("\nResultados de los clasificadores:")
    print(f"XGBoost: '{prediccion_xgb_2}'")
    print(f"LightGBM: '{prediccion_lgb_2}'")
    print(f"CatBoost: '{prediccion_cat_2}'")

    promedio_probs_severo = (probs_xgb_severo + probs_lgb_severo + probs_cat_severo) / 3.0
    prediccion_voting_severo = np.argmax(promedio_probs_severo, axis=1)[0]
    prediccion_severo = class_names_severo[prediccion_voting_severo]

    return f"Resultado Modelo 2: La imagen es clasificada como Grado {prediccion_severo}"
```

Nota. Código en Python sobre la construcción del ensamble de clasificadores para los modelos.

3.6.7. ANÁLISIS DE ROBUSTEZ Y ESTABILIDAD DEL MODELO PROPUESTO

3.6.7.1. PROCEDIMIENTO DE ROBUSTEZ

Para asegurar la fiabilidad de los resultados y mitigar el impacto de la aleatoriedad inherente en procesos de entrenamiento como la inicialización de pesos y la división de datos, se ejecutó un total de 10 repeticiones del

entrenamiento para los modelos binario y de severo, incluyendo la evaluación del ensamble (Deza, Jonnalagadda, & Eckstein, 2017).

Figura 31

Lista de robustez en modelos entrenados



Nota. Se repitió la ejecución del modelo propuesto para garantizar su robustez.

3.6.7.2. ANÁLISIS ESTADÍSTICO DEL MODELO

A partir de los 10 resultados de la métrica Accuracy Global, se calcularon la media y la desviación estándar. El rendimiento final del sistema se reporta en el formato promedio ± desviación estándar, proporcionando una medida robusta, fiable de la precisión y estabilidad del modelo propuesto.

Figura 32

Código de cálculo de media y desviación estándar

```
num_ejecuciones_1 = len(lista_de_accuracies_xgb)
num_ejecuciones_2 = len(lista_de_accuracies_lgb)
num_ejecuciones_3 = len(lista_de_accuracies_cat)

accuracies_array_1 = np.array(lista_de_accuracies_xgb)
mean_accuracy_1 = np.mean(accuracies_array_1)
std_accuracy_1 = np.std(accuracies_array_1)

accuracies_array_2 = np.array(lista_de_accuracies_lgb)
mean_accuracy_2 = np.mean(accuracies_array_2)
std_accuracy_2 = np.std(accuracies_array_2)

accuracies_array_3 = np.array(lista_de_accuracies_cat)
mean_accuracy_3 = np.mean(accuracies_array_3)
std_accuracy_3 = np.std(accuracies_array_3)
```

Nota. Código en Python sobre el cálculo de la media y desviación estándar de los porcentajes de la métrica de Exactitud global.

3.6.8. DISEÑO DEL MODELO DE PREDICCIÓN FINAL

En esta última fase metodológica, se cargaron todos los componentes en solo un modelo de predicción. Este nuevo modelo resultante combina de manera articulada los extractores de características basados en CNN, los clasificadores de Gradient Boosting y la lógica de ensamble por votación. Donde es capaz de recibir una radiografía, clasificarla inicialmente como no severo o severo; de ser severo, se deriva el caso al modelo severo para hallar el grado específico.

Figura 33

Código de carga de componentes

```
class_names_binario = ['no_severo', 'severo']
class_names_severo = ['2', '3', '4']

extractor_binario = construir_extractor_binario()
extractor_binario.load_weights('modelo_binario_extractor_weights.h5')
print("Modelo 1 (Binario) cargado")

extractor_severo = construir_extractor_severo()
extractor_severo.load_weights('modelo_severo_extractor_weights.h5')
print("Modelo 2 (Severo) cargado")

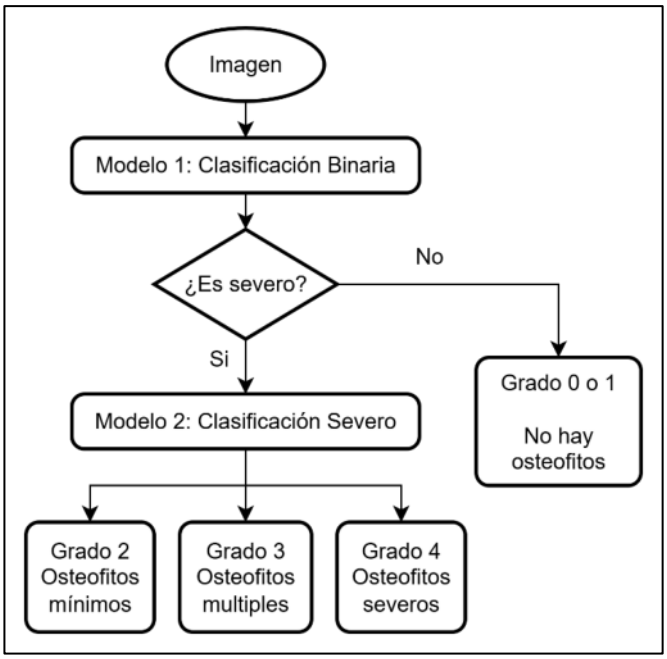
xgb_binario = joblib.load('modelo_binario_xgb.joblib')
lgb_binario = joblib.load('modelo_binario_lgb.joblib')
cat_binario = joblib.load('modelo_binario_cat.joblib')
print("Clasificadores del Modelo 1 (Binario) cargados")

xgb_severo = joblib.load('modelo_severo_xgb.joblib')
lgb_severo = joblib.load('modelo_severo_lgb.joblib')
cat_severo = joblib.load('modelo_severo_cat.joblib')
print("Clasificadores del Modelo 2 (Severo) cargados")
```

Nota. Código en Python sobre la combinación de los modelos, pesos, extractores y clasificadores.

Figura 34

Diagrama de flujo del modelo de predicción final



Nota. Muestra el flujo de clasificación, comenzando con un modelo binario para evaluar la severidad de la imagen, de ser necesario, identificar el grado severo de los osteofitos.

4. RESULTADOS Y DISCUSIÓN

4.1. RENDIMIENTO DEL MODELO DE REFERENCIA

En la fase inicial de la investigación, se entrenó un modelo de CNN para la clasificación directa de las 5 clases del grado de osteofitos. Como se observa en la Tabla 5, este modelo alcanzó una precisión global de 53% como máximo, evidenciando las limitaciones de este enfoque.

Tabla 5

Resultados de los modelos CNN iniciales

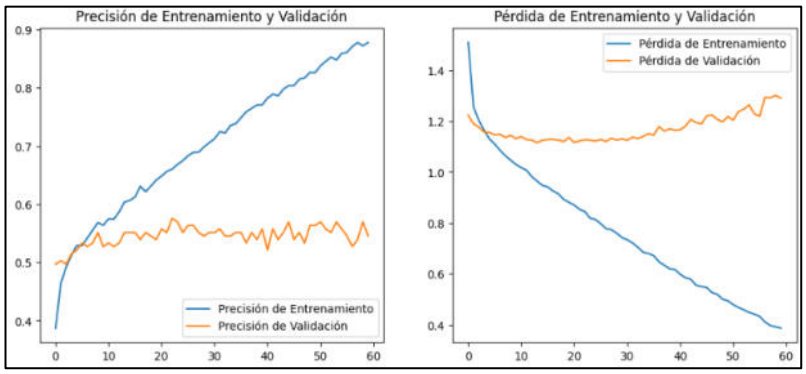
Modelo	Epochs	Average	Overfitting
ResNet50	60	0.52	Alta
ResNet50	27	0.49	Leve
EfficientNetB1	50	0.48	Leve
EfficientNetB1 + XGB	19	0.53	Nula

Nota. Se usó inicialmente modelos CNN para la clasificación directa de las 5 clases existentes.

Después de analizar las precisiones de entrenamientos y validaciones de los modelos de la Tabla 5, se observa casos de sobreajuste, también conocido como Overfitting. A pesar de haber alcanzado un resultado máximo del 53%, su capacidad de generalización o memorización no es óptima, lo que lleva a descartar el enfoque de 5 clases como una opción viable para la investigación.

Figura 35

Monitoreo del entrenamiento con modelo ResNet50

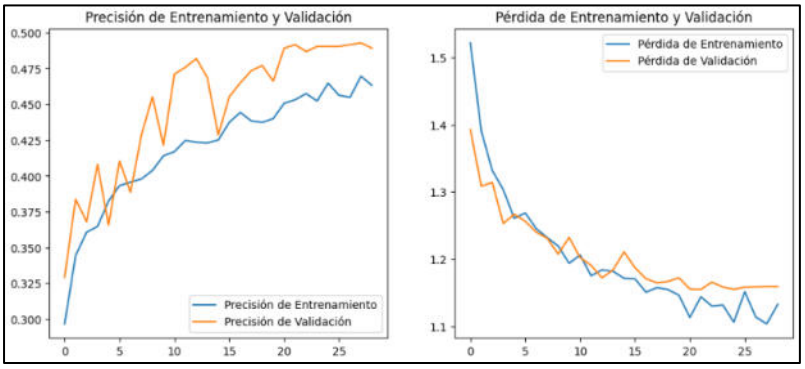


Nota. Se muestra que el entrenamiento mejora progresivamente, mientras que la validación presenta fluctuaciones.

Posteriormente, se observó que el modelo utilizando EfficientNetB1 manejó el sobreajuste de manera más efectiva, aunque proporcionó un valor similar al anterior. Sin embargo, de acuerdo con la matriz de confusión, todavía existen dificultades para clasificar las clases minoritarias. Por lo que aún se puede ajustar dicho modelo para obtener mayores resultados.

Figura 36

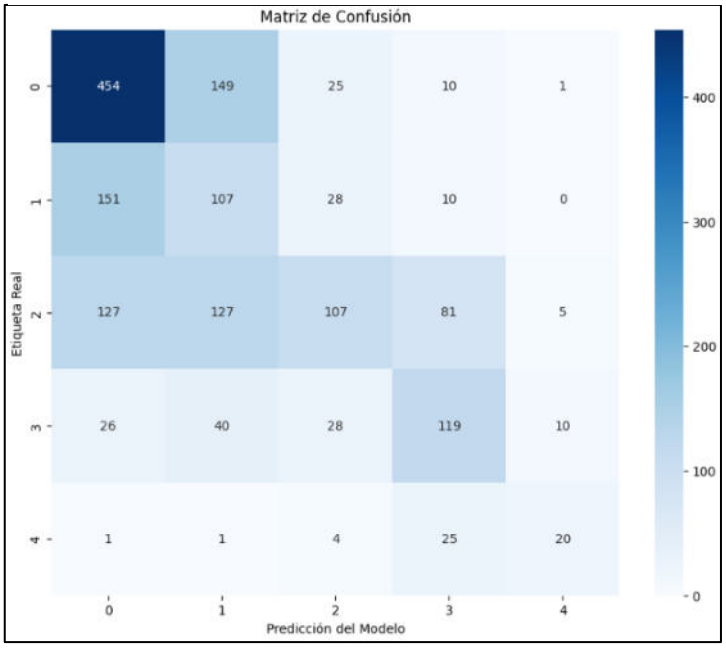
Monitoreo del entrenamiento con modelo EfficientNetB1



Nota. Se muestra que el entrenamiento y la validación mejora progresivamente hasta cierto punto.

Figura 37

Matriz de confusión del entrenamiento con EfficientNetB1



Nota. Se muestra la exactitud de las clases y el tipo de errores cometidos en cada clase.

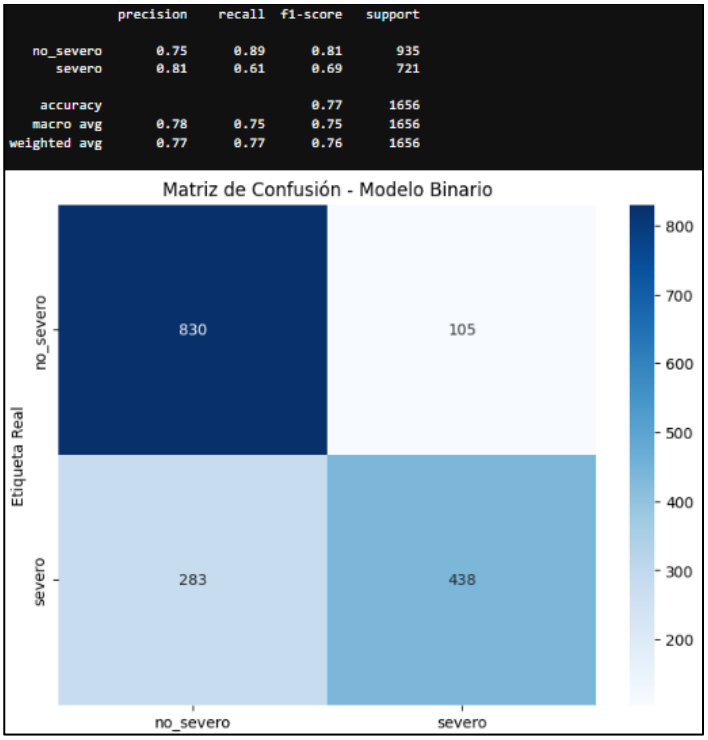
4.2. RESULTADOS DE LA CLASIFICACIÓN JERÁRQUICA

4.2.1. RENDIMIENTO INICIAL DEL MODELO 1 (BINARIO)

El primer componente de esta estrategia, el clasificador binario, fue evaluado en la tarea de distinguir casos “No severos” de “Severos”. Inicialmente, el modelo alcanzó una precisión global del 77%. En la Figura 38 muestra el reporte de clasificación y la matriz de confusión mostrando un alto Recall de 0.89 para la clase “No severo”. En comparación con la idea de usar las 5 clases directamente.

Figura 38

Reporte de clasificación y matriz de confusión del entrenamiento inicial con modelo binario



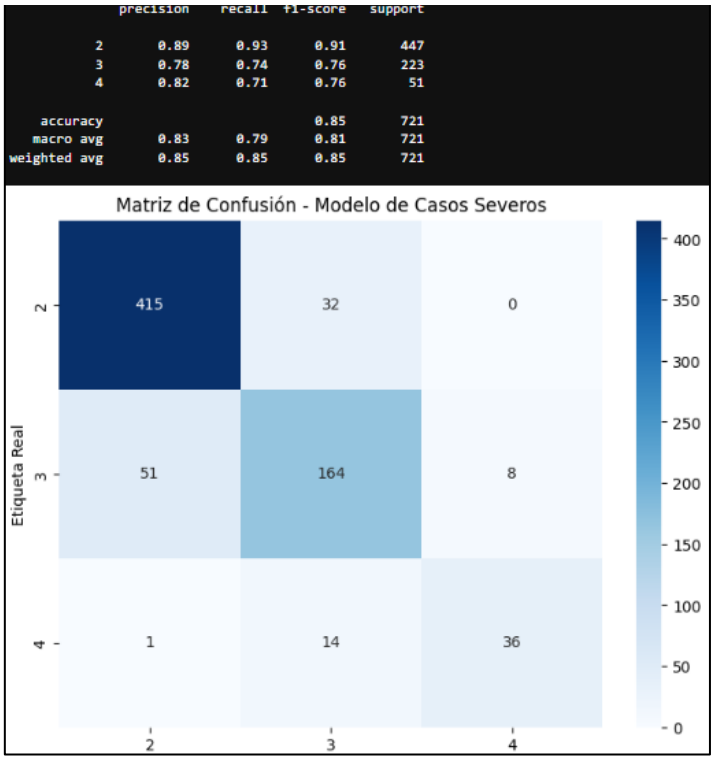
Nota. Se muestra mejoría en las métricas de evaluación y la matriz de confusión del modelo binario, mostrando Precision, Recall y F1-Score para ambas clases.

4.2.2. RENDIMIENTO INICIAL DEL MODELO 2 (SEVERO)

El segundo componente, el modelo severo, se evaluó en la tarea de clasificar los casos graves. Inicialmente, logró una precisión general del 85%. Como se detalla en la Figura 39, el modelo demostró una alta efectividad para la clase '2'. A pesar de la variación en las imágenes dentro del conjunto de datos "Test", el modelo distingue con éxito entre las clases.

Figura 39

Reporte de clasificación y matriz de confusión del entrenamiento inicial con modelo severo



Nota. Se muestra mejoría en las métricas de evaluación y la matriz de confusión del modelo severo, mostrando Precision, Recall y F1-Score para todas las clases.

4.3. RESULTADOS DE LOS MODELOS HÍBRIDOS

4.3.1. COMPARACIÓN DE ARQUITECTURAS CNN

Para optimizar el extractor de características, se evaluó el desempeño de siete arquitecturas de redes neuronales convolucionales (CNN). La Tabla 6 presenta un resumen de los resultados obtenidos por los modelos base, destacando a EfficientNetB0, por lograr la más alta exactitud entre 0.85 y 0.89 en las tareas de clasificación binaria y

severa. Cabe señalar que es la arquitectura más sencilla de esta familia, EfficientNetB0, logrando clasificar con efectividad frente a la de sus versiones más complejas, las cuales poseen un mayor número de parámetros y capas. Por lo tanto, considerando la optimización del tiempo y los recursos, EfficientNetB0 resulta adecuado para abordar el problema.

Tabla 6

Resultados máximos de todos los modelos CNN aplicados

Modelo	Accuracy CNN		Accuracy Fine-Tuning	
	Binario	Severo	Binario	Severo
ResNet50	0.59	0.39	0.65	0.49
DenseNet121	0.52	0.49	0.67	0.55
InceptionV3	0.62	0.52	0.66	0.62
EfficientNetB0	0.74	0.72	0.84	0.88
EfficientNetB1	0.73	0.70	0.84	0.88
EfficientNetB2	0.71	0.69	0.83	0.88
EfficientNetB3	0.68	0.69	0.84	0.88

Nota. Se muestran los valores máximos de exactitud obtenidos por cada modelo, antes y después del Fine-Tuning.

4.3.2. COMPARACIÓN DE CLASIFICADORES DE GRADIENT BOOSTING

En la etapa final de evaluación de modelos, se implementaron tres clasificadores avanzados basados en Gradient Boosting: XGBoost, LightGBM y CatBoost. El análisis comparativo de su rendimiento, detallado en la Tabla 7, demuestra que los modelos basados en ResNet50, DenseNet121 e InceptionV3 tuvieron un rendimiento

moderado y consistente entre los tres clasificadores, con métricas que oscilaron entre 0.61-0.70 para la clase binaria y 0.62-0.66 para la clase severo. Sin embargo, se observa un salto significativo al emplear las arquitecturas de la familia EfficientNet como extractores de características.

Tabla 7

Resultados máximos de los clasificadores de Gradient Boosting

Modelo	XGBoost		LightGBM		CatBoost	
	Binario	Severo	Binario	Severo	Binario	Severo
ResNet50	0.68	0.61	0.69	0.63	0.69	0.62
DenseNet121	0.70	0.63	0.70	0.63	0.70	0.63
InceptionV3	0.68	0.65	0.68	0.66	0.68	0.66
EfficientNetB0	0.84	0.89	0.85	0.89	0.85	0.89
EfficientNetB1	0.85	0.88	0.85	0.89	0.84	0.89
EfficientNetB2	0.83	0.88	0.84	0.89	0.84	0.89
EfficientNetB3	0.84	0.88	0.84	0.88	0.85	0.88

Nota. Se muestran los valores máximos obtenidos con clasificadores XGBoost, LightGBM y CatBoost.

Destaca especialmente el modelo EfficientNetB0, que confirma que esta arquitectura no solo supera ampliamente a otras, sino que además replica su alto desempeño de manera robusta e independiente del clasificador utilizado. Si bien EfficientNetB1, B2 y B3 mostraron un rendimiento similar, la elección de la versión B0 se justifica plenamente al ofrecer la mejor relación entre precisión, eficiencia computacional y número de parámetros. Los resultados obtenidos demuestran que la combinación de EfficientNetB0 con

clasificadores de Gradient Boosting constituye la estrategia óptima para el problema abordado en esta investigación.

4.4. RENDIMIENTO DE MÉTRICAS Y ETIQUETAS

La Tabla 8 presenta el desempeño comparativo de los tres algoritmos de Gradient Boosting en el modelo binario. Los clasificadores demostraron un rendimiento sólido y consistente, con valores promedios de exactitud global del 0.85 en los tres modelos. Para la clase "No Severo", los algoritmos exhibieron en Precision (0.85), Recall (0.89), F1-Score (0.87). La clase "Severo" mostró valores de Precision ligeramente inferiores (0.84) y mayor variabilidad en Recall (0.79-0.89) y F1-Score (0.82-0.87).

Tabla *8*

Resultados máximos de reportes de clasificación del modelo binario

Métricas	XGBoost		LightGBM		CatBoost	
	No Severo	Severo	No Severo	Severo	No Severo	Severo
Precision	0.84	0.84	0.85	0.85	0.85	0.84
Recall	0.88	0.79	0.89	0.79	0.89	0.79
F1-Score	0.86	0.81	0.87	0.82	0.87	0.82
Accuracy	0.84		0.85		0.85	

Nota. Se muestran las métricas de evaluación para cada clase, utilizando los clasificadores XGBoost, LightGBM y CatBoost.

Todos los clasificadores se desempeñan mejor en identificar casos "No Severo", con diferencias de 0.05 en F1-Score y 0.01 en Precision.

La Tabla 9 presenta el desempeño comparativo de los tres clasificadores de Gradient Boosting mediante sus matrices de confusión. Donde LightGBM mostró mejores resultados en la clase "No Severo", mientras que CatBoost obtuvo el mayor rendimiento en la clase "Severo".

Tabla 9

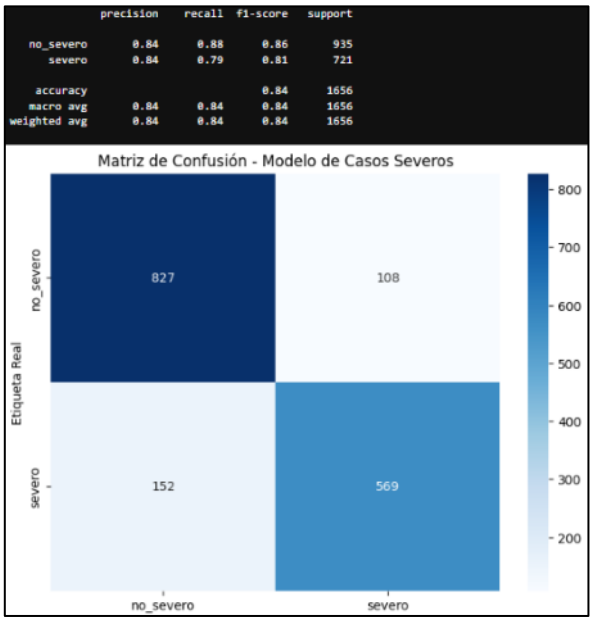
Resultados máximos de las matrices de confusión del modelo binario

Etiquetas	XGBoost		LightGBM		CatBoost	
	No Severo	Severo	No Severo	Severo	No Severo	Severo
No Severo	827	108	836	99	828	107
Severo	152	569	151	570	149	572

Nota. Se muestran los valores máximos de casos clasificados correcta y erróneamente mediante matrices de confusión.

Figura 40

Reporte de clasificación y matriz de confusión del entrenamiento ideal con modelo binario



Nota. Se muestra los valores de la evaluación del modelo binario mediante el reporte de clasificación y matriz de confusión.

La Tabla 10 presenta el desempeño de los clasificadores de Gradient Boosting en el modelo severo (Grado 2, 3 y 4). Los tres clasificadores demostraron exactitud global uniforme de 0.89, evidenciando consistencia en la capacidad predictiva general. Para la clase 2, los modelos alcanzaron: Precision entre 0.91-0.92, Recall de 0.94-0.95 y F1-Score de 0.93, indicando desempeño excepcional en la identificación de este grado. La clase 3 mostró valores inferiores con precisión de 0.83, Recall entre 0.80-0.81 y F1-score de 0.81-0.82, reflejando mayor desafío en la clasificación de esta categoría intermedia.

Tabla 10

Resultados máximos de reportes de clasificación del modelo severo

Métricas	XGBoost			LightGBM			CatBoost		
	2	3	4	2	3	4	2	3	4
Precision	0.92	0.83	0.86	0.91	0.83	0.88	0.92	0.83	0.84
Recall	0.94	0.81	0.75	0.95	0.80	0.75	0.94	0.81	0.75
F1-Score	0.93	0.82	0.80	0.93	0.81	0.81	0.93	0.82	0.79
Accuracy	0.89			0.89			0.89		

Nota. Se muestran las métricas de evaluación para cada clase, utilizando los clasificadores XGBoost, LightGBM y CatBoost.

Se observa una diferencia en el rendimiento entre clases, con diferencias de aproximadamente 0.10 puntos en las métricas F1 entre la clase 2 y las clases 3-4. Esta variación sugiere mayor complejidad en la distinción de grados avanzados de severidad, posiblemente debido a características visuales entre categorías adyacentes.

La Tabla 11 presenta el desempeño de los clasificadores, destacándose por la distinción de grados extremos. XGBoost destacó en detectar la clase 3, LightGBM mostró el mejor equilibrio global en todas las clases, y CatBoost presentó mayor capacidad de detección de la clase 4.

Tabla 11

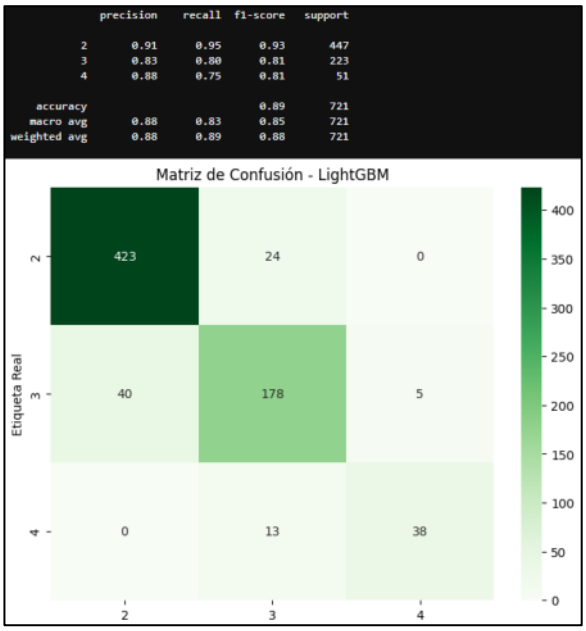
Resultados máximos de las matrices de confusión del modelo severo

Etiquetas	XGBoost			LightGBM			CatBoost		
	2	3	4	2	3	4	2	3	4
2	422	25	0	423	24	0	421	26	0
3	37	180	6	40	178	5	36	180	7
4	0	13	38	0	13	38	1	12	38

Nota. Se muestran los valores máximos de casos clasificados correcta y erróneamente mediante matrices de confusión.

Figura 41

Reporte de clasificación y matriz de confusión del entrenamiento ideal con modelo severo



Nota. Se muestra los valores de la evaluación del modelo binario mediante el reporte de clasificación y matriz de confusión.

4.5. ANÁLISIS DE ERRORES Y CASOS MAL CLASIFICADOS

Para complementar la evaluación del rendimiento de los modelos, se realizó un análisis detallando las clasificaciones erróneas para identificar patrones en los errores de predicción y las posibles razones subyacentes.

- Identificación de clases incorrectas: Las métricas de rendimiento y las matrices de confusión de los modelos, según la Tabla 10, indican que las clases de Grados 3 (Moderado) y Grado 4 (Severo) fueron las más difíciles de clasificar. El Grado 3 tuvo el menor valor en Precision (0.83), el Grado 4 tuvo en Recall (0.75) y en F1-score ambos grados fueron inferiores frente al Grado 2. En la Figura 39, detalla que los casos reales de Grado 3 eran a menudo clasificados erróneamente como Grado 2 o Grado 4, y viceversa para el Grado 4.

- Ejemplos visuales de clasificación incorrecta: Para una comprensión más detallada de estos errores, se examinaron visualmente radiografías que fueron mal clasificadas. Se identificaron casos donde el modelo predijo un grado de osteofitos cuando el real era diferente, especialmente en las transiciones entre grados adyacentes.

Figura 42

Diagnóstico de Grado 2 en una radiografía de Grado 3



Nota. Se muestra un error de diagnóstico (Clase 2).

Figura 43

Diagnóstico de Grado 4 en una radiografía de Grado 3



Nota. Se muestra un error de diagnóstico (Clase 4).

- Causas de los errores de clasificación: Las inconsistencias y errores de clasificación observados pueden atribuirse a varios factores como la similitud visual entre grados adyacentes. La alta similitud visual entre grados adyacentes de osteofitos, como la distinción entre Grado 2 y Grado 3, o Grado 3 y Grado 4, presenta un desafío para cualquier modelo de clasificación basado en imágenes. Las diferencias morfológicas pueden ser sutiles y minuciosas incluso para la interpretación humana, lo que se traslada a la dificultad del modelo para establecer fronteras de decisión claras.

Figura 44

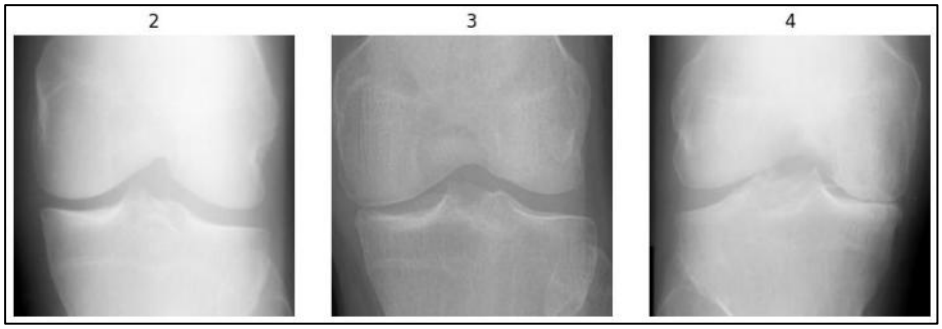
Muestras de radiografías de los 5 Grados de Osteofitos



Nota. Se muestran radiografías clasificadas en los grados 0 al 4 según la presencia y severidad de osteofitos.

Figura 45

Muestras de radiografías de Grados 2, 3 y 4



Nota. Se muestran imágenes de los grados más avanzados de osteofitos para facilitar su diferenciación visual.

4.6. EVALUACIÓN FINAL DE LOS MODELOS

La estrategia de clasificación jerárquica demostró su efectividad en la mayoría de los casos, en dos escenarios clínicos representativos. Para el primer escenario, la imagen de la radiografía “9088841L.png” mostró un diagnóstico unánime entre los tres clasificadores (XGBoost, LightGBM, CatBoost) con predicción 'no_severo', permitiendo finalizar el diagnóstico eficientemente en la primera etapa como "No Severo (Grado 0 o 1)". Este resultado ejemplifica la capacidad de filtrado del modelo binario, evitando procesamiento innecesario en casos evidentes.

Figura 46

Resultado del diagnóstico final del modelo binario

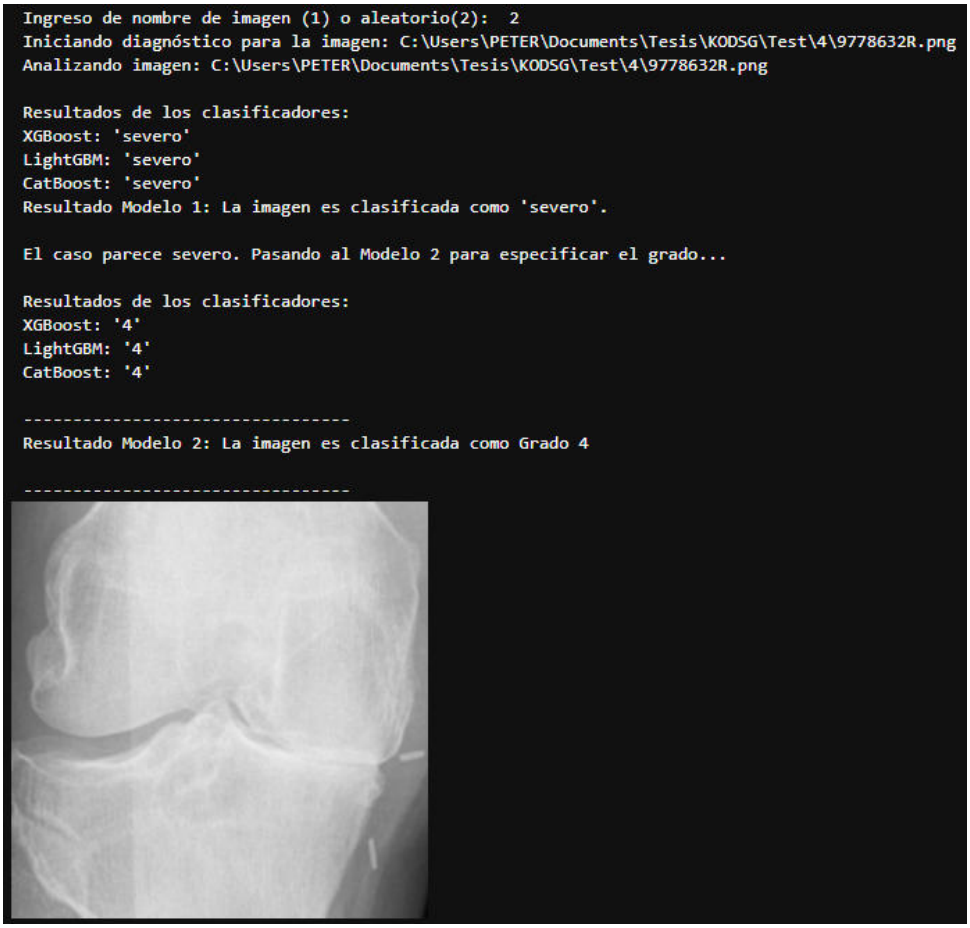


Nota. Se muestra el resultado del diagnóstico realizado por los clasificadores XGBoost, LightGBM y CatBoost, indicando que la imagen fue clasificada como “no severo”.

En otro caso, la imagen de la radiografía “97786328.png” activó el segundo escenario al obtener una predicción unánime de 'severo' en todos los clasificadores. El modelo severo mostró consistencia diagnóstica absoluta, clasificando concordantemente el caso como "Grado 4" en los tres algoritmos. Este caso demuestra la precisión del sistema en grados avanzados y la efectividad de la arquitectura en cascada para manejar casos complejos que requieren análisis multiclases.

Figura 47

Resultado del diagnóstico final del modelo binario y severo



Nota. Se muestra el resultado del diagnóstico realizado por los clasificadores XGBoost, LightGBM y CatBoost, indicando que la imagen fue clasificada como “severo” y luego se especifica como “Grado 4”.

4.7. RESULTADOS DEL ANÁLISIS DE ROBUSTEZ

Para asegurar la estabilidad del modelo propuesto, se llevó a cabo el proceso de entrenamiento y evaluación en su totalidad en 10 ocasiones. El rendimiento final del sistema, expresado como media ± desviación estándar, fue el siguiente:

- Modelo Binario: 84.3% ± 0.46%
- Modelo Severo: 88.6% ± 0.49%.

Esta métrica de precisión global respalda la robustez y la reproducibilidad de los resultados, donde el clasificador CatBoost en el modelo binario apunta a ser el mejor, mientras que, en el modelo severo, el clasificador LightGBM ha demostrado una ligera ventaja sobre los demás.

Tabla 12

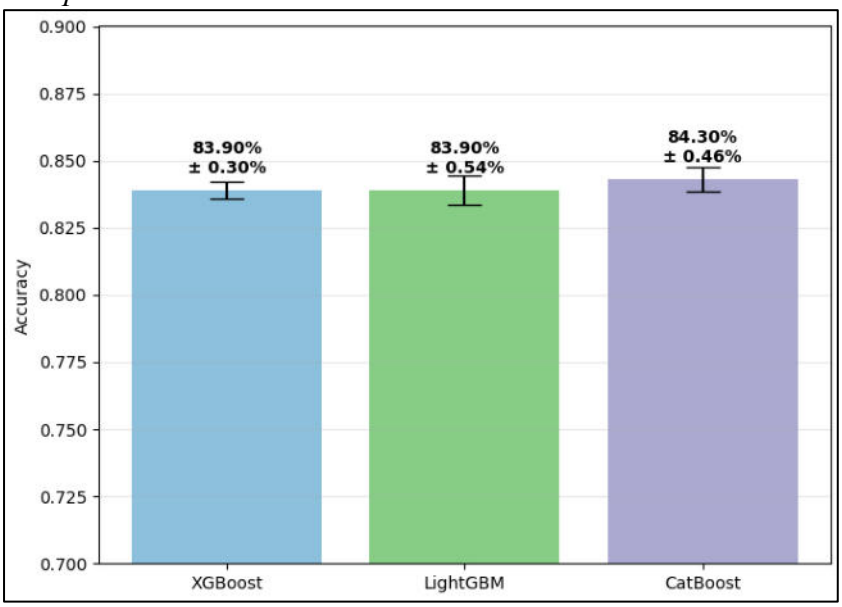
Resultados del análisis de robustez de los modelos propuestos

Ejecución de robustez	EfficientNetB0					
	XGBoost		LightGBM		CatBoost	
	Binario	Severo	Binario	Severo	Binario	Severo
1	0.84	0.88	0.84	0.89	0.84	0.89
2	0.84	0.89	0.84	0.89	0.84	0.88
3	0.84	0.89	0.84	0.89	0.84	0.88
4	0.84	0.88	0.84	0.89	0.85	0.88
5	0.84	0.89	0.85	0.89	0.85	0.89
6	0.84	0.88	0.84	0.88	0.84	0.88
7	0.83	0.89	0.83	0.88	0.84	0.88
8	0.84	0.89	0.83	0.88	0.84	0.88
9	0.84	0.88	0.84	0.88	0.84	0.88
10	0.84	0.88	0.84	0.89	0.85	0.89

Nota. Se muestran los resultados de precisión obtenidos en distintas ejecuciones para evaluar la estabilidad de los modelos propuestos.

Figura 48

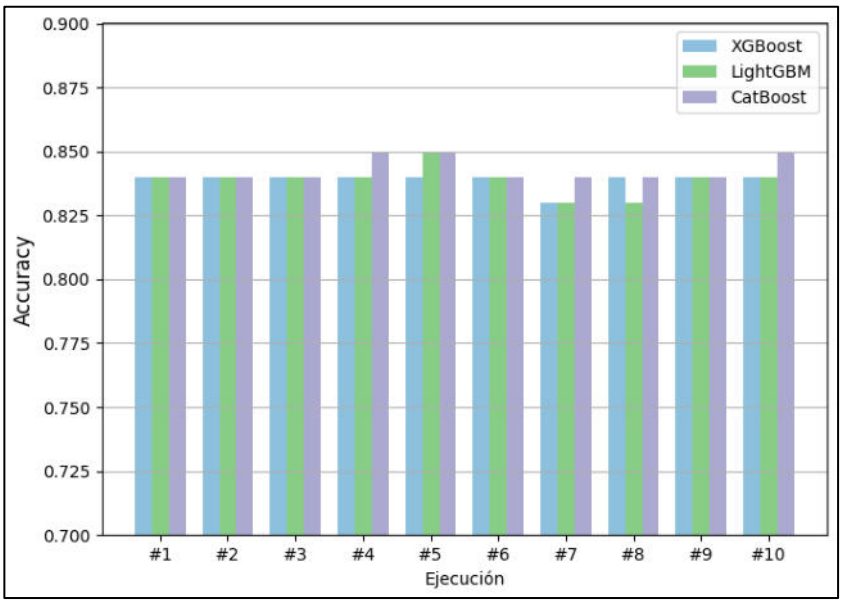
Comparación de robustez del modelo binario



Nota. Se muestra la variabilidad en el rendimiento del modelo binario, evaluando su estabilidad de resultados.

Figura 49

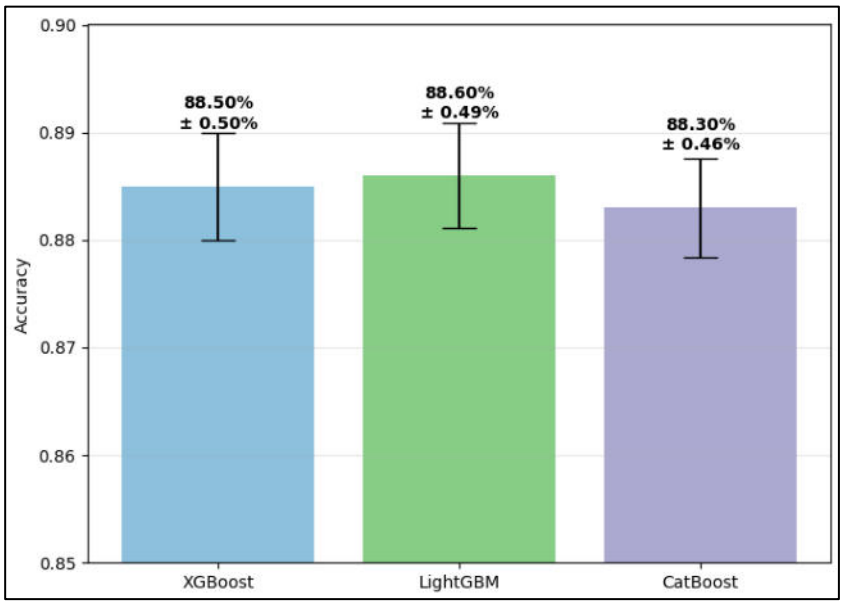
Comparación de ejecuciones del modelo binario



Nota. Se muestran los resultados obtenidos en distintas ejecuciones del modelo binario, permitiendo observar la consistencia en su desempeño.

Figura 50

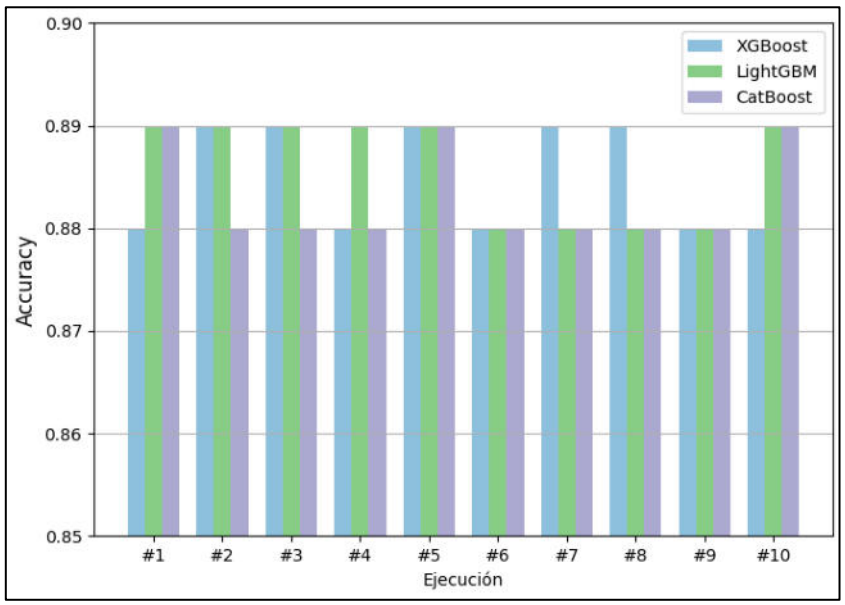
Comparación de robustez del modelo severo



Nota. Se muestra la variabilidad en el rendimiento del modelo severo, evaluando su estabilidad de resultados.

Figura 51

Comparación de ejecuciones del modelo severo



Nota. Se muestran los resultados obtenidos en distintas ejecuciones del modelo severo, permitiendo observar la consistencia en su desempeño.

4.8. COMPARACIÓN CON EL ESTADO DEL ARTE

Para cumplir con el objetivo específico de evaluar y comparar el modelo propuesto con los enfoques existentes en la literatura, se recopilaron investigaciones recientes que abordan el mismo problema de clasificación. La Tabla 13 presenta una comparación en términos de arquitectura, recursos computacionales, estrategias de preprocesamiento y precisión obtenida. Se observa que, si bien modelos como el de Bose et al. (2024) reportan un accuracy de 98%, la búsqueda de soluciones óptimas no se detiene solo en esta métrica. La constancia de investigaciones muestra que la comunidad científica reconoce los factores del problema, donde la eficiencia computacional y la escalabilidad en entornos reales son criterios tan cruciales como la precisión pura. En este contexto, el modelo híbrido propuesto se posiciona de manera competitiva con sus ventajas clave:

- Optimización de recursos: Uso de arquitecturas ligeras (EfficientNet) contra modelos pesados (VGG, ensambles complejos, generación de imágenes, etc.).
- Flexibilidad: Capacidad de trabajar con diferentes agrupaciones de clases ((0,1/2,3,4) y (2,3,4)) sin requerir reentrenamiento desde cero.

Esta comparación demuestra que el modelo propuesto no solo cumple con estándares de precisión, sino que lo hace de manera eficiente y adaptable, siendo una solución viable para entornos clínicos con restricciones tecnológicas.

Tabla 13

Comparación de modelos del estado del arte

Términos	Bose et al. (2024)	Devarapaga et al. (2024)	Segura et al. (2024)	Mohammed et al. (2023)	Momenpour et al. (2025)	Negron (2025)
Modelo	VGG	SCAE Net, VGG16, DenseNet	Conv Next	VGG, ResNet, MobileNet, DenseNet	Efficient Net B5	Efficient Net B0
Clasificación jerárquica	Si	~	~	Si	~	Si
Épocas	~	+50	+15	+80	+50	+50
Tiempo	~	~	~	~	1 Hr	1.3 Hr
Memoria	Alto	Alto	Alto	Alto	Bajo	Bajo
Recursos gráficos	~	~	Landing Lens	12GB	11GB	6GB
Aumentación de datos	~	~	~	~	Si	Si
Muestreo	Si	~	~	~	~	Si
Flexibilidad	~	~	~	~	~	Si
Preservación de datos	~	~	~	~	Si	Si
Generación virtual	~	~	~	~	Si	Si
Generación física	Si	~	~	~	~	~
Callbacks	~	~	~	Si	Si	Si
Regularizador	~	Si	~		Si	Si
Extractor de características	Si	Si	Si	Si	Si	Si
Clasificador GB	~	~	~	~	~	Si
Ensamble	~	~	~	~	~	Si
Accuracy	0.98	0.96	0.95	0.83	0.82	0.89

Nota. Hr = Horas, GB = Gigabytes. Tabla elaborada por el autor con datos adaptados de los estudios de Bose et al. (2024), Devarapaga et al. (2024), Segura et al. (2024), Mohammed et al. (2023), Momenpour et al. (2025). Las referencias completas de cada estudio se encuentran en la bibliografía.

4.9. LIMITACIONES DEL ESTUDIO

A pesar de los resultados prometedores y el enfoque del modelo híbrido propuesto, el estudio presenta varias limitaciones que deben ser reconocidas. Estas limitaciones muestran el alcance actual de la investigación y proporcionan vías claras para el trabajo futuro:

- Tamaño y desbalance del Dataset: El estudio fue limitado por un desbalance de clases expuesto en la Tabla 1. Las clases avanzadas de osteofitos, Grado 3 (Moderado) y Grado 4 (Severo), estaban desequilibradas con 757 y 173 imágenes, respectivamente. A pesar de aplicar técnicas como el cálculo de pesos y sobremuestreo SMOTE, clasificar estas clases siguió siendo difícil, afectando la capacidad de generalización de los modelos.

- Ausencia de generación física de imágenes: El estudio abordó el aumento de datos y técnicas de sobremuestreo, en tiempo real, evitando métodos computacionalmente intensivos de generación física de imágenes sintéticas a gran escala, como los modelos del estado del arte, Bose et al. (2024), Momenpour et al. (2025) y Mohammed et al. (2023). Esta decisión, donde se priorizó la eficiencia y la preservación de los datos originales, limitó la diversidad sintética de las muestras, restringiendo potencialmente la robustez del entrenamiento y el rendimiento en clases minoritarias.

- Relación entre exactitud y eficiencia computacional: El modelo propuesto, basado en EfficientNetB0, prioriza la eficiencia computacional, alcanzando una exactitud del 89% que lo hace viable para entornos con recursos limitados. No obstante, esta relación entre eficiencia y rendimiento resulta en una exactitud inferior al estado del arte, como el de Bose et al. (2024), posicionándolo como una solución competitiva en aplicaciones que valoran más la operatividad que la máxima precisión diagnóstica.

- Exploración limitada de arquitecturas CNN: Aunque se realizó un estudio comparativo de arquitecturas CNN y se seleccionó EfficientNetB0 por su balance óptimo entre rendimiento y eficiencia, la investigación sugiere la exploración de otras arquitecturas para tareas de visión por computadora. Esto sugiere que, si bien la arquitectura elegida se ajusta a las limitaciones del proyecto, podría haber otras que logren un rendimiento aún mayor.

- Estado del estudio para trabajos futuros: El estudio se presenta como la implementación de un modelo funcional y una metodología validada que sienta las bases para el futuro desarrollo de herramientas de apoyo al diagnóstico clínico de osteofitos. Esto implica que el modelo actual no es una solución clínica completamente integrada, sino una etapa fundamental que requiere más desarrollo y validación clínica para su aplicación práctica.

5. CONCLUSIONES

5.1. CONCLUSIÓN GENERAL

En esta investigación se propuso desarrollar e implementar un modelo de aprendizaje automático para el diagnóstico del grado de osteofitos en radiografías. Donde se ha llegado a la siguiente conclusión general.

Se cumplió exitosamente el objetivo general a través de un modelo híbrido funcional para el diagnóstico del grado de osteofitos en radiografías. Donde la estrategia más eficaz no era un solo modelo de CNN ni un solo clasificador GB, sino una clasificación jerárquica que utiliza de manera combinada un modelo de red neuronal convolucional para la extracción automática de características visuales de las radiografías y un clasificador de gradient boosting para la toma de decisiones final. Mediante un método de ensamble se pudo designar a votación entre los mejores modelos hallar el veredicto del diagnóstico, aprovechando la robustez de los modelos propuestos. Dicho método es capaz de tomar una radiografía y emitir una decisión para el diagnóstico del grado de severidad del osteofito, validando así la viabilidad del enfoque híbrido propuesto.

5.2. CONCLUSIONES ESPECÍFICAS

Respecto al primer objetivo específico. Se logró diseñar y validar una arquitectura de aprendizaje automático con un rendimiento optimizado para la complejidad inherente del problema. Se concluye que un único modelo

que intenta clasificar las cinco clases de forma simultánea presenta limitaciones significativas, debido principalmente a la alta similitud visual entre grados adyacentes y al desbalance de clases. La arquitectura diseñada, por el contrario, descompone el problema de forma jerárquica, demostrando ser una solución superior. El diseño final consiste en:

- Un modelo 1 (binario), que actúa como un clasificador de alta fiabilidad para diferenciar casos "no severos" de "severos".
- Un modelo 2 (severo), que se enfoca únicamente en los casos severos para especificar el grado (2, 3 o 4), manejando esta tarea más sutil con mayor habilidad que un modelo original de cinco clases.

Esta arquitectura de dos etapas representa ser óptima, no solo por alcanzar una alta precisión diagnóstica de hasta 89%, sino también por su eficiencia computacional. A diferencia de muchos modelos del estado del arte que dependen de arquitecturas considerablemente pesadas y recursos de servidor (data center), el modelo propuesto logra un rendimiento competitivo en hardware de consumo (laptop comercial), sin requerir la generación de imágenes offline ni tiempos de ejecución exorbitantes.

Respecto al segundo objetivo específico. Se cumplió con el objetivo mediante una evaluación y comparación sistemática de las diferentes arquitecturas desarrolladas durante la investigación. El modelo jerárquico híbrido final fue comparado cuantitativamente con el modelo de referencia del estado del arte, que es el enfoque de Deep Learning de extremo a extremo (una CNN única para las 5 clases). La evaluación, basada en

métricas como el F1-score por clase y el análisis de la matriz de confusión, demostró de manera concluyente la superioridad del sistema jerárquico propuesto. Mientras que el modelo de 5 clases fallaba en distinguir clases, el sistema jerárquico logró, primero, una separación robusta de casos leves y severos y, segundo, una clasificación más enfocada de los grados de alta severidad.

Durante la evaluación se demostró que el modelo propuesto es significativamente mejor en términos de eficiencia y potencial de despliegue que muchos enfoques del estado del arte. Mientras que otros estudios se centran en modelos monolíticos de gran tamaño para obtener ganancias marginales de precisión, esta clasificación jerárquica alcanza un rendimiento de alto nivel (89%) con una fracción de recursos computacionales con respecto al estado del arte. Esta eficiencia lo convierte en una solución no solo precisa, sino también práctica y accesible para su posible integración en entornos clínicos con recursos limitados.

En definitiva, esta tesis no solo resulta en la implementación de un modelo funcional, sino que también aporta una metodología validada que aborda eficazmente los desafíos de la clasificación de imágenes médicas, sentando las bases para el desarrollo futuro de herramientas de apoyo al diagnóstico clínico.

6. RECOMENDACIONES

Tras el desarrollo y evaluación del modelo propuesto, se han reconocido diversas maneras para mejorar este trabajo. A continuación, se presentan las siguientes recomendaciones:

- Incrementar el tamaño del dataset: Se recomienda la recopilación de un mayor número de radiografías, especialmente para las clases minoritarias (Grado 3 y 4). Datos más balanceados permitirán mejorar el rendimiento y la capacidad de generalización del Modelo 2.

- Enriquecimiento con datos clínicos: Se recomienda incorporar datos no visuales, como la edad, sexo, índice de masa corporal e historial clínico. La combinación de las características extraídas con estos datos tabulares podría mejorar la precisión y dar mayor relevancia clínica.

- Refinamiento del clasificador jerárquico: El Modelo 1 agrupa las clases 0 y 1 como "No Severo". Se recomienda entrenar un tercer modelo que se encargue solo de diferenciar entre el Grado 0 y el Grado 1, generando un clasificador de radiografías sanas. Además, de un modelo donde pueda descartar clases que no sean subyacentes, como el Grado 0, 2, 4.

- Exploración de otras arquitecturas de CNN: Los modelos de la familia EfficientNet indicaron ser un extractor de características eficaz, se recomienda usar otras arquitecturas óptimas en cuanto al uso de recursos para tareas de visión por computadora y comparar los resultados.

- Validación clínica con expertos: Se recomienda realizar un estudio en el que las predicciones del sistema sean comparadas con las evaluaciones de un panel de radiólogos expertos. Esto permitiría no solo validar la precisión del modelo frente a un estándar humano, sino también evaluar la concordancia inter-observador y entender en qué tipo de casos el modelo difiere de la opinión experta.

- Desarrollo de una interfaz de usuario (UI): Para facilitar su uso por parte de personal no técnico, se recomienda encapsular el pipeline final en una aplicación de escritorio o web simple, donde el usuario pueda cargar una radiografía y recibir el diagnóstico de forma clara e intuitiva.

7. REFERENCIAS

Mayo Clinic. (2022). *Espolones óseos - Síntomas y causas*. https://www.mayoclinic.org/es-es/diseases-conditions/bone-spurs/symptoms-causes/syc-20370212

De la Torre, E. (2021). *Lesiones de disco: degeneración con osteofitos*. Neurocirugía de la Torre. https://www.neurocirugiaequipodelatorre.es/lesiones-de-disco-degeneracion-con osteofitos#:%7E:text=El%20especialista%20puede%20detectar%20la,tomograf%C3%ADa%20axial%20computarizada%20(TAC).

M.I.N.S.A. (2019). *Se estima que en el Perú cada año se diagnostican más de 100 casos nuevos de artritis reumatoidea*. [Noticia]. Ministerio de Salud - Gobierno del Perú. https://www.gob.pe/institucion/minsa/noticias/27840-se-estima-que-en-el-peru-cada-ano-se-diagnostican-mas-de-100-casos-nuevos-de-artritis-reumatoidea

Aenolastname, A. (2022). *Perú: Lima registra 300 casos anuales de artritis infantil*. AméricaEconomía. https://www.americaeconomia.com/sector-publico/peru-lima-registra-300-casos-anuales-de-artritis-infantil

Kokkotis, C., Ntakolia, C., Moustakidis, S., Giakas, G., & Tsaopoulos, D. (2022). *Explainable machine learning for knee osteoarthritis diagnosis based on a novel fuzzy feature selection methodology. Physical and Engineering Sciences in Medicine*, *45(1),* 219-229.

Tiulpin, A., Thevenot, J., Rahtu, E., Lehenkari, P., & Saarakkala, S. (2018). *Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach. Scientific reports, 8(1),* 1727.

Antony, J., McGuinness, K., O'Connor, N. E., & Moran, K. (2016, December). *Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In 2016 23rd international conference on pattern recognition (ICPR) (pp. 1195-1200).* IEEE.

Wang, Y., Wang, X., Gao, T., Du, L., & Liu, W. (2021). *An automatic knee osteoarthritis diagnosis method based on deep learning: Data from the Osteoarthritis Initiative*. *Journal of Healthcare Engineering*, *2021*, Article ID 5586529.

Tycowicz, C. (2020). *Towards shape-based knee osteoarthritis classification using graph convolutional networks*. En *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (pp. 750–753). IEEE.

Durán Olivar, D. (2022). *Automatic Detection of Knee Joints and Classification of Knee Osteoarthritis Severity from Plain Radiographs using CNNs* (Tesis de maestría, Universitat Oberta de Catalunya).

Yang, X., Tang, P., Zou, K., Dai, D. (2024). *HCGN: Hierarchical Convolution and Graph Network for Predicting Knee Osteoarthritis. 2024 IEEE International Conference on Medical Artificial Intelligence (MedAI).*

Bengani, V. (2024). *Hybrid Learning Systems: Integrating Traditional Machine Learning with Deep Learning Techniques* [Tesis de maestría, Jain University].

Mavaie, P., Holder, L., & Skinner, M. K. (2023). *Hybrid deep learning approach to improve classification of low-volume high-dimensional data. BMC Bioinformatics, 24(419).*

Chieregato, M., Frangiamore, F., Morassi, M., Baresi, C., Nici, S., Bassetti, C., Bnà, C., & Galelli, M. (2022). *A hybrid machine learning/ deep learning COVID-19 severity predictive model from CT images and clinical data. Scientific Reports, 12, 7890.*

Fisioterapia Online. (s.f.). *Osteofitos*. https://www.fisioterapia-online.com/glosario/osteofitos

Fisiolution. (2023). *Qué son y cómo afectan a las articulaciones del cuerpo* [Fotografía]. https://fisiolution.com/osteofitos-que-son-y-como-afectan-a-las-articulaciones-del-cuerpo/

Mayo Clinic. (2020). *Osteoartritis – Síntomas y causas*. https://www.mayoclinic.org/es/diseases-conditions/osteoarthritis/symptoms-causes/syc-20351925

Mayo Clinic. (2023). *Artritis.* Mayo Clinic. https://www.mayoclinic.org/es/diseases-conditions/arthritis/symptoms-causes/syc/20350772

Enciclopedia Médica A.D.A.M. [Internet]. Atlanta (GA): A.D.A.M., Inc.; ©1997–2025. *Imagenología y radiología*. https://medlineplus.gov/spanish/ency/article/007451.htm

Top Doctors. (s.f.). *Radiografía*. Diccionario Médico Top Doctors. https://www.topdoctors.es/diccionario-medico/radiografia

IBM. (s.f.). *Machine learning*. IBM Think. https://www.ibm.com/think/topics/machine-learning

Built In. (s.f.). *Deep learning*. https://builtin.com/machine-learning/deep-learning

Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. *Neural networks*, *61*, 85-117.

Sancho, F. (s.f.). *Redes neuronales*. Universidad de Sevilla. https://www.cs.us.es/~fsancho/Blog/posts/Redes_Neuronales.md.html

Analytics Vidhya. (2022). *Basics of CNN in deep learning*. https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/

Analytics Vidhya. (2020). *Convolutional neural networks (CNN)*. Medium. https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-a78e78c1ba94

DataScientest. (2023). *Convolutional neural network: Everything you need to know*. https://datascientest.com/en/convolutional-neural-network-everything-you-need-to-know

KnightHawkK. (2020). *How to use Conv2D layers as fully connected layers.* Medium. https://medium.com/@knighthawkk/how-to-use-conv2d-layers-as-fully-connected-layers-b0a82eb8a408

Cortez, O. (2020). *Introducción a los métodos de ensamble y al algoritmo de XGBoost: Caso práctico*. Medium. https://medium.com/@oscars.cortezmo/introducción-a-los-métodos-de-ensamble-y-al-algoritmo-de-xgboost-caso-práctico-e8cb0d58394b

Akif, M. A., Butun, I., Williams, A., & Mahgoub, I. (2025). *Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset. arXiv preprint arXiv:2502.12382.*

Mukhlif, A. A., Al-Khateeb, B., & Mohammed, M. A. (2023). *Incorporating a Novel Dual Transfer Learning Approach for Medical Images. Sensors, 23(2), 570.*

Paitán, H. Ñ., Dueñas, M. R. V., Vilela, J. J. P., & Delgado, H. E. R. (2019). *Metodología de la investigación cuantitativa-cualitativa y redacción de la tesis*. Ediciones de la U.

Silla, Carlos & Freitas, Alex. (2011). *A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery. 22. 31-72.*

ShashwatWork. (2022). *Knee osteoarthritis dataset with severity grading* [Dataset]. Kaggle. https://www.kaggle.com/datasets/shashwatwork/knee-osteoarthritis-dataset-with-severity/data

Yang, Y., Zhang, L., Du, M., Bo, J., Liu, H., Ren, L., Li, X., & Deen, M. J. (2021). *A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions. Computers in Biology and Medicine, 139*, 104887.

Viso AI. (2023). *EfficientNet: How to achieve state-of-the-art performance with compound scaling*. https://viso.ai/deep-learning/efficientnet/

Deza, A., Jonnalagadda, A., & Eckstein, M. (2017). *Towards metamerism via foveated style transfer. arXiv preprint arXiv:1705.10041.*

Bose, A. S. C., Srinivasan, C., & Joy, S. I. (2024). *Optimized feature selection for enhanced accuracy in knee osteoarthritis detection and severity*

classification with machine learning. Biomedical Signal Processing and Control, 97, 106670.

Devarapaga, S., & Thumma, R. (2024). *Knee Osteoarthritis SCAENet: Adaptive Knee Osteoarthritis Severity Assessment Using Spatial Separable Convolution with Attention-Based Ensemble Networks with Hybrid Optimization Strategy. Journal of Imaging Informatics in Medicine, 38*, 1563–1580.

Segura, F. M., Segura, F. P., Zudaire, M. P. L., & Segura, F. V. (2024). *Advances in Artificial Intelligence for automated knee osteoarthritis classification using the IKDC system. European Journal of Orthopaedic Surgery & Traumatology, 35, 32.*

Mohammed, A. S., Hasanaath, A. A., Latif, G., & Bashar, A. (2023). *Knee Osteoarthritis Detection and Severity Classification Using Residual Neural Networks on Preprocessed X-ray Images. Diagnostics, 13(8),* 1380.

Momenpour, T., & Abu Mallouh, A. (2025). *Optimizing CNN-Based Diagnosis of Knee Osteoarthritis: Enhancing Model Accuracy with CleanLab Relabeling. Diagnostics, 15(11),* 1332.

8. ANEXOS

8.1. MATRIZ DE CONSISTENCIA

PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLE	DIMENSIÓN	INDICADOR	ESCALA
GENERAL			Independiente: Modelo Híbrido de Machine Learning.	CNN + GB	Image Input Size Base Model Callbacks Optimizer Epochs Batch-size Dropout Learning Rate L2 Regularization Activation Function Loss Function Classifier	[224 x 224] [CNN's] [ES, RLR] [Adam, SWA] [30-500] [32] [0.4 – 0.7] [1e-3 – 1e-10] [1e-2 – 1e-4] [sigmoid – softmax] [Binary C., Categorical C.] [GB's]
¿Puede un modelo híbrido de Machine Learning diagnosticar diferentes grados de osteofitos en radiografías?	Implementar un modelo híbrido de Machine Learning para diagnosticar diferentes grados de osteofitos en radiografías.	Con un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías.				
ESPECÍFICOS			Dependiente: Diagnóstico del grado osteofito en radiografías.	Osteofitos	Accuracy Precision Recall F1-score	[0 - 1] [0 - 1] [0 - 1] [0 - 1]
¿Con una arquitectura óptima de un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías?	Diseñar una arquitectura óptima de un modelo híbrido de Machine Learning para diagnosticar diferentes grados de osteofitos en radiografías.	Con una arquitectura óptima de un modelo híbrido de Machine Learning se puede diagnosticar diferentes grados de osteofitos en radiografías.				

¿Es el modelo propuesto significativamente mejor que los modelos del estado del arte para diagnosticar diferentes grados de osteofitos en radiografías?	Evaluar y comparar los modelos del estado del arte con el modelo propuesto para diagnosticar diferentes grados de osteofitos en radiografías.	Con respecto a los modelos del estado del arte, el modelo propuesto tiene rendimiento significativo para diagnosticar diferentes grados de osteofitos en radiografías.				

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos	: Peter Gianfranco Negron Yucra
DNI/Carnet de Extranjería/Pasaporte N°	: 73220492
Correo electrónico	: PETER.FRKL17@GMAIL.COM
Teléfono/Celular	: 972020754

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	X			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (X)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (X)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : Diagnóstico del grado de osteofito en radiografías mediante un modelo híbrido de Machine Learning.

b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : 2.02.04

c) Fecha de publicación : 13 de Octubre de 2025

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : Hugo Euler Tito Chura
b) DNI/Pasaporte/Cédula : 01315850
c) ORCID : 0000-0001-8269-0655

FIRMA :

DOCUMENTO DE IDENTIDAD : 73220492
NOMBRES Y APELLIDOS : Peter Gianfranco Negron Yucra

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(X) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 73220492

NOMBRES Y APELLIDOS : Peter Gianfranco Negron Yucra

FECHA : 13 de Octubre de 2025



ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

Yo, Peter Gianfranco Negron Yucra ,
de la Escuela Profesional de:
Ingeniería de Sistemas e Informática ,
de la Universidad Nacional de Moquegua, identificado(a) con DNI N°... 73220492 , autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:

Diagnóstico del grado de osteofito en radiografías mediante un modelo híbrido de Machine Learning

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 13 de Octubre de 2025


Huella digital

Nombres y Apellidos: Peter Gianfranco Negron Yucra
DNI: 73220492

Hugo Euler TITO CHURA

TESIS 2025 v2-INFORME IA.pdf

Informe Final Tesis observaciones levantadas - Peter Negrón

Tesis 2025

Universidad Nacional de Moquegua

Detalles del documento

Identificador de la entrega
trn:oid:::1:3337302325

Fecha de entrega
12 sep 2025, 12:31 p.m. GMT-5

Fecha de descarga
12 sep 2025, 12:39 p.m. GMT-5

Nombre del archivo
TESIS_2025_v2-INFORME_IA.pdf

Tamaño del archivo
2.1 MB

98 páginas

13.709 palabras

83.421 caracteres

9% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 8 palabras)

Fuentes principales

8% Fuentes de Internet
3% Publicaciones
4% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Fuentes principales

8% Fuentes de Internet
3% Publicaciones
4% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1 Internet
medium.com <1%

2 Internet
www.ibm.com <1%

3 Internet
hdl.handle.net <1%

4 Internet
www.coursehero.com <1%

5 Trabajos del estudiante
Universidad Internacional de la Rioja <1%

6 Trabajos del estudiante
Universidad Continental <1%

7 Internet
repositorio.unam.edu.pe <1%

8 Internet
www.mdpi.com <1%

9 Internet
oa.upm.es <1%

10 Trabajos del estudiante
Universidad Nacional de Colombia <1%

11 Internet
repositorio.uncp.edu.pe <1%

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 818-2025-CU

Moquegua, 20 de noviembre de 2025

VISTOS:

El Proveído N° 4717-2025-R de fecha 14 de noviembre de 2025 emitido por el Rectorado, Informe N° D0480-2025-UNAM-R-SG de fecha 14 de noviembre de 2025 emitido por el Secretario General, Informe Técnico N° D0160-2025-SEGE-GT emitido por la Especialista Administrativo I–Grados y Títulos, el Acuerdo del Pleno del Consejo Universitario de Sesión Extraordinaria N° 44-2025-SE-CU-UNAM, de fecha 19 de noviembre de 2025, y;

CONSIDERANDO:

Que, conforme al 4to párrafo del artículo 18, de la Constitución Política del Perú, dispone que cada Universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico. Y el artículo 8° de la Ley Nº 30220-Ley Universitaria, establece que la referida autonomía inherente a las Universidades, se ejerce de conformidad con la Constitución y demás normativa aplicable. Concordante con el artículo 1, de la Ley Nº 31520, Ley que restablece la autonomía y la institucionalidad de las universidades peruanas, así como la Ley N° 32141, Ley que modifica la Ley N° 30220-Ley Universitaria, a fin de fortalecer la institucionalidad de las Universidades Públicas licenciadas;

Que, el Artículo 58, de la Ley N° 30220-Ley Universitaria, establece que el Consejo Universitario es el máximo órgano de gestión, dirección y ejecución académica y administrativa de la Universidad. Concordante con el artículo 18° del Estatuto de la Universidad Nacional de Moquegua, aprobado bajo Resolución de Asamblea Universitaria N° 015-2025-AU-UNAM;

Que, el artículo 59 de la Ley Universitaria-Ley N° 30220, establece las atribuciones del Consejo Universitario, y específicamente el numeral 59.9, establece; Conferir los grados académicos y los títulos profesionales aprobados por las Facultades y Escuela de Posgrado, así como otorgar distinciones honoríficas y reconocer y revalidar los estudios, grados y títulos de universidades extranjeras, cuando la universidad está autorizada por la Superintendencia Nacional de Educación Superior Universitaria. Concordante con el numeral 14, del artículo 20° del Estatuto de la Universidad Nacional de Moquegua;

Que, el Sr. Rector es el personero y representante legal de la Universidad Nacional de Moquegua, conforme a lo dispuesto por la Ley Nº 30220-Ley Universitaria, tiene a su cargo la Dirección, Conducción y Gestión del Gobierno Universitario en todos sus ámbitos, concordante con el artículo 24° y numeral 2 del artículo 26° del Estatuto de la Universidad Nacional de Moquegua;

Que, mediante Informe Técnico N° D0160-2025-SEGE-GT de fecha 13 de noviembre de 2025, e Informe N° D0480-2025-UNAM-R-SG de fecha 14 de noviembre de 2025, el Secretario General de la UNAM, remite al Rectorado el expediente para el trámite de Título Profesional de Ingeniero de Sistemas e Informática del Bachiller: NEGRON YUCRA PETER GIANFRANCO de la Escuela Profesional de Ingeniería de Sistemas e Informática, quien se encuentra apto para la expedición del Título Profesional de Ingeniero de Sistemas e Informática, al haber cumplido satisfactoriamente con los requisitos legales y reglamentarios, exigidos en el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, conforme lo señala la Dirección de Actividades y Servicios Académicos y la Dirección de la Escuela Profesional de Ingeniería de Sistemas e Informática, razón por la cual, el señor Secretario General incluye dicho expediente como punto de agenda para ser tratado en Sesión Extraordinaria de Consejo Universitario de fecha 19 de noviembre de 2025;

Que, el artículo 44° de la Ley N° 30220, Ley Universitaria señala que las universidades otorgan los grados académicos de Bachiller, Maestro, Doctor y los Títulos Profesionales que correspondan, a nombre de la Nación (...). Asimismo, la Ley N° 31803, modifica el artículo en mención adicionando; La Superintendencia Nacional de Educación Superior Universitaria (SUNEDU) determina los criterios, así como establece y supervisa el procedimiento administrativo para el reconocimiento de grados y títulos otorgados en el extranjero, en el marco de la normativa vigente;

Que, el numeral 45.2 del artículo 45° de la Ley Nº 30220, Ley Universitaria, precisa los requisitos mínimos para la obtención del título profesional, como son: el grado de Bachiller y la aprobación de una tesis o trabajo de suficiencia profesional. Asimismo, establece que el título profesional solo se puede obtener en la universidad en la cual se haya obtenido el grado de bachiller;

Que, mediante Resolución de Comisión Organizadora N° 1153-2021-UNAM, se aprobó el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, el mismo que fue modificado a través de las Resoluciones de Comisión Organizadora N° 410; 453 y 890-2022, N° 880-2023-UNAM y N° 081, 484-2024-UNAM el cual establece los requisitos y norma los procedimientos para la obtención de los grados académicos y títulos profesionales conferidos por esta Casa Superior de Estudios;

Que, bajo el amparo de los considerandos precedentes, el Pleno del Consejo Universitario en Sesión Extraordinaria N° 44-2025-SE-CU-UNAM, de fecha 19 de noviembre de 2025, emitió, el siguiente, Acuerdo, POR UNANIMIDAD CONFERIR, a nombre de la Nación el Título Profesional de Ingeniero de Sistemas e Informática por la modalidad de sustentación de tesis al Bachiller: NEGRON

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 818-2025-CU

Moquegua, 20 de noviembre de 2025

YUCRA PETER GIANFRANCO, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingenierías de la Universidad Nacional de Moquegua;

En uso de las facultades y atribuciones conferidas por el Art. 18, de la Constitución Política del Perú, la Ley N° 30220 Ley Universitaria, el Estatuto de la Universidad Nacional de Moquegua, y a lo acordado por el pleno del Consejo Universitario:

SE RESUELVE:

ARTÍCULO 1°: CONFERIR, A NOMBRE DE LA NACIÓN al Bachiller

- **NEGRON YUCRA PETER GIANFRANCO**

TESIS : "DIAGNÓSTICO DEL GRADO DE OSTEOFITO EN RADIOGRAFÍAS MEDIANTE UN MODELO HÍBRIDO DE MACHINE LEARNING", cuyo Asesor Principal fue el Dr. Hugo Euler Tito Chura.

El **TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS E INFORMÁTICA.**

Por la modalidad de SUSTENTACIÓN DE TESIS, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingenierías de la Universidad Nacional de Moquegua.

ARTÍCULO 2°: ENCARGAR, al Secretario General a través de quien corresponda, para que realice las acciones pertinentes relacionadas con la inscripción correspondiente en el Registro de Grados y Títulos de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

ARTÍCULO 3°: DISPONER, el cumplimiento de la presente Resolución a todas las dependencias de la Universidad Nacional de Moquegua.

ARTÍCULO 4°: DISPONER la publicación de la presente resolución en el Portal Institucional de la Universidad Nacional de Moquegua www.unam.edu.pe

REGÍSTRESE, PUBLÍQUESE Y CÚMPLASE;

UNIVERSIDAD NACIONAL DE MOQUEGUA SECRETARIA GENERAL

ABOG. EDGAR PINTO HUANCAVILCA
SECRETARIO GENERAL

UNIVERSIDAD NACIONAL DE MOQUEGUA CONSEJO UNIVERSITARIO

DR. HUGO RUBEN MARCA MAQUERA
RECTOR
PRESIDENTE DEL CONSEJO UNIVERSITARIO

DISTRIBUCION:
VRA
VRI
FAIN
EPIS
GyT
DASA
OTI
Interesado
c.c. Arch.

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe